

IMPLEMENTASI ALGORITMA NEGASCOUT PADA PERMAINAN NINE MEN'S MORRIS

Dany Kurniawan¹
dany.ndlah0@gmail.com

Rosa Delima²
rosadelima@ukdw.ac.id

Nugroho Agus H³
cnuq@ukdw.ac.id

Abstract

Application development nowadays many are using technology/knowledge of Artificial Intelligence study. This research tries to make an intelligence agent that can play Nine Men's Morris game using Negascout algorithm. The purpose of this research are to compare the effectiveness of Negascout algorithm in exploring the best move along with time that required, compared with Alpha-Beta Pruning algorithm. The result from this research are Negascout algorithm explores fewer nodes and require less time in the search of the best move compared with Alpha-Beta Pruning algorithm.

Keywords: *artificial intelligence, Negascout, Nine Men's Morris.*

1. Pendahuluan

Perkembangan teknologi dewasa ini berkembang dengan sangat cepat. Perkembangan aplikasi banyak menggunakan teknologi / ilmu dalam bidang kedercadsan buatan atau *Artificial Intelligence* (AI). Aplikasi yang paling menarik dan begitu berkembang secara luas yaitu aplikasi AI dalam permainan. Penerapan AI dalam dunia permainan pada umumnya adalah digunakan untuk membuat sebuah agen yang dapat mengambil keputusan terhadap kondisi permainan yang dinamis, berdasarkan pengetahuan dan aturan-aturan yang ditanamkan.

Nine Men's Morris merupakan salah satu permainan berjenis *board game* klasik yang di mainkan oleh dua pemain, dengan 24 titik dimana bidak diletakkan. Tiap pemain memiliki 9 keping bidak, yang biasanya terdiri dari bidak warna putih dan hitam. Permainan ini terdiri dari 3 fase, yaitu fase pembukaan, pertengahan, dan akhir. Permainan berakhir apabila: 1. Pemain yang pertama kali memiliki bidak kurang dari 3 akan kalah, 2. Pemain yang tidak bisa menjalankan bidak pada gilirannya akan kalah, 3. Jika kondisi permainan berulang terus menerus, permainan dianggap seri.

Pengimplementasian agen cerdas dalam permainan yang berjenis *board game* biasanya menggunakan algoritma *optimal desicions* (mencari solusi optimal dari berbagai jalan yang berurutan), seperti *minimax*, *negamax*, dan *alpha-beta pruning*. Seiring dengan berjalannya waktu, muncullah algoritma baru, yaitu algoritma *negascout*. Algoritma *negascout* merupakan pengembangan dari algoritma *scout* dikombinasikan dengan algoritma *alpha-beta*. Dalam penelitian ini, penulis ingin mencoba mengimplementasikan algoritma *negascout* dan metode evaluasi ke dalam bentuk aplikasi *Nine Men's Morris*.

¹Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana.

²Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana.

³Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana.

2. Rumusan dan Batasan Masalah

Berdasarkan latar belakang dari penulisan tugas akhir ini, maka disusunlah bahasan rumusan masalah. Adapun masalah yang akan dibahas dalam laporan ini adalah apakah algoritma yang digunakan lebih efektif (dilihat dari segi waktu, jumlah node, dan presentase kemenangan) dalam hal mengeksplorasi node-node. Keungkinan langkah dibandingkan algoritma *Alpha-Beta Pruning*.

Ruang Lingkup penelitian ini akan menerapkan beberapa parameter-parameter batasan dalam sistem ini, yaitu :

- a. Algoritma yang digunakan adalah algoritma *negascout*.
- b. Menggunakan aturan permainan *Nine Men's Morris*.

3. Implementasi Sistem

3.1 Sistem Penggambaran Kondisi Papan

Kondisi awal papan permainan digambarkan dengan peletakkan dua puluh empat titik yang saling berhubungan dan membentuk papan permainan dengan kondisi awal tanpa bidak (kosong) pada papan permainan. Kondisi awal ini sendiri merupakan standar dari permainan *Nine Men's Morris*.

Tiap pemain akan memiliki bidak masing-masing 9. Papan ini sendiri digambarkan secara statis di awal permainan berdasarkan sebuah *array* yang dinyatakan sebagai sumber informasi penggambaran papan awal (Gambar 3.6, Bab 3). Dari *array* inilah nantinya papan akan dibaca ulang dan digambarkan pada halaman antarmuka permainan. Cara penggambarannya sendiri adalah dengan melakukan perulangan pada tiap indek *array* dan melihat nilai didalamnya, apakah itu berupa titik atau kosong. Perulangan akan dilakukan dari baris dan kolom pertama *array* hingga baris dan kolom akhir *array* menggunakan *looping* dalam *looping* sebanyak 7 kali. Selanjutnya tiap baris dan kolom pada *array* akan dicek apakah berisi nilai yang menandakan titik. Jika ditemukan nilai titik, maka sistem akan membuat objek "Titik". Kumpulan objek Titik tersebut dikelola dalam suatu *array*. Selanjutnya kumpulan objek "Titik" tersebut disusun membentuk papan permainan menggunakan fungsi "draw".

3.2 Sistem Pencarian Langkah

Tiap giliran sebelum pemain mengambil langkah akan diawali dengan pencarian langkah oleh sistem permainan. Pencarian langkah pada agen cerdas akan disertai dengan nilai dari langkah tersebut yang akan menjadi perhitungan untuk langkah terbaik yang akan diambil. Proses pencarian langkah ini sendiri dibagi ke dalam 3 fase sesuai dengan alur permainan.

a. Fase 1

Proses pencarian langkah pada fase 1 dilakukan dengan melakukan perulangan sebanyak 24 kali sesuai dengan jumlah titik pada papan permainan. Di dalam perulangan kemudian dilakukan pengecekan terhadap tiap titik.

Kriteria pengecekannya sendiri adalah titik tersebut dicek apakah titik tersebut merupakan titik yang masih kosong. Setiap titik yang kosong tersebut diisi oleh bidak yang bersangkutan dan menjadi sebuah langkah. Langkah-langkah tersebut kemudian akan dikelola ke dalam sebuah *array* "moves".

b. Fase 2

Proses pencarian langkah pada fase 2 dilakukan dengan melakukan perulangan sebanyak 24 kali sesuai dengan jumlah titik pada papan permainan. Di dalam perulangan kemudian dilakukan pengecekan terhadap tiap titik.

Kriteria pengecekannya sendiri adalah titik tersebut dicek apakah titik tersebut berisi bidak pemain yang bersangkutan. Setiap titik tersebut akan dicek adakah titik tetangganya yang merupakan titik kosong yang sesuai dengan jalur pada papan permainan. Setiap titik kosong tersebut menjadi sebuah langkah. Langkah-langkah tersebut kemudian akan dikelola ke dalam sebuah *array* "moves".

c. Fase 3

Proses pencarian langkah pada fase 3 dilakukan dengan melakukan perulangan sebanyak 24 kali sesuai dengan jumlah titik pada papan permainan. Di dalam perulangan kemudian dilakukan pengecekan terhadap tiap titik.

Kriteria pengecekannya sendiri adalah titik tersebut dicek apakah titik tersebut berisi bidak pemain yang bersangkutan. Bidak tersebut dapat dipindahkan ke titik lain yang merupakan titik kosong dan menjadi sebuah langkah. Langkah-langkah tersebut kemudian akan dikelola ke dalam sebuah *array* "moves".

3.3 Sistem Pengambilan Langkah Terbaik Oleh Agen Cerdas

Fungsi pencarian langkah pada sistem ini adalah penerapan algoritma *Negascout* dalam pencarian terbaik. Berikut penjelasan mengenai fungsi *Negascout* pada sistem ini :

- a. Fungsi ini sendiri merupakan fungsi rekursif yang mana memiliki *stopper* pada baris 4 pada gambar. *Stopper* ini bekerja saat fungsi mencapai kedalaman yang telah ditentukan atau permainan selesai.
- b. Jika kondisi *stopper* belum terpenuhi maka dilakukan pencarian *successor* atau kemungkinan langkah yang ada dalam kedalaman tersebut.
- c. Tiap *successor* akan dicek *nodenya* menggunakan looping sebanyak total *node* dalam *successor*.
- d. Pengecekan *node* sendiri menggunakan fungsi *negascout* kembali untuk mencari kemungkinan-kemungkinan langkah dari *node* tersebut.
- e. Jika pengembalian nilai dari fungsi *negascout* lebih besar dari induknya, maka akan dilakukan *update* nilai.
- f. Dilakukan pengecekan pada *node* cabang pertama dalam pencarian. Pada algoritma *Negascout*, cabang pertama dalam *tree* dianggap sebagai nilai terbaik.
- g. Akan dilakukan pencarian nilai terbaik dari *node* apabila pengecekan pada langkah f tidak terpenuhi.
- h. Jika suatu *node* ternyata lebih kecil dari nilai terbaik yang telah didapatkan, maka semua *node* dengan kedalaman pada cabang yang bersangkutan akan diabaikan pencariannya.
- i. Dilakukan *update* dari nilai *adaptiveBeta* yang menjadi tanda bahwa pencarian berada pada cabang pertama.
- j. Fungsi ini akan mengembalikan hasil nilai pencarian berdasarkan fungsi "evaluasi".

3.4 Sistem Memakan Bidak Lawan

Sistem memakan bidak lawan adalah sistem yang dijalankan saat pemain memakan bidak lawan. Pemain dapat memakan bidak lawan apabila pemain dapat membentuk *mill/morris*. Setelah pemain dapat membentuk *mill/morris* pemain dipersilahkan untuk memilih satu bidak lawan yang akan dibuang dari permainan. Mekanisme sistem memakan bidak lawan ini adalah dengan mengecek tiap langkah apakah langkah tersebut menghasilkan *mill/morris*. Apabila terpenuhi maka pemain yang bersangkutan mendapat langkah tambahan yaitu langkah untuk memilih bidak mana yang akan dibuang.

4. Analisis Sistem

4.1 Hasil Pengujian Pengguna Melawan Negascout

Pengujian ini dilakukan untuk mengetahui tingkat kemenangan yang dapat diraih oleh agen *Negascout* melawan pengguna. Tingkat kedalaman yang dipakai adalah 3. Dalam pengujian ini lima pengujian pertama pengguna memainkan bidak hitam, selanjutnya bidak putih.

Dalam pengujian ini hasil yang disajikan pada bagian agen cerdas *Negascout* merupakan hasil pembulatan perhitungan rata-rata tiap langkah pada suatu pertandingan. Pada tiap langkah (*move*) sistem menghitung berapa jumlah *node* yang dikunjungi dan berapa waktu yang dibutuhkan untuk menentukan langkah terbaik (dalam satuan milidetik).

Dari hasil pengujian terlihat bahwa user memiliki pelunag yang lebih tinggi jika memilih bidak putih, yaitu 4 kali menang dari 5 pertandingan atau sekitar 80% dibandingkan jika memilih bidak hitam, yaitu 2 kali menang dari 5 pertandingan atau sekitar 40%.

4.2 Hasil Perbandingan Negascout dengan Alpha Beta Pruning

Pengujian ini dilakukan pada kedalaman 2 sampai 5. Hasil pengujian merupakan hasil pembulatan perhitungan rata-rata tiap langkah pada suatu pertandingan.

Dalam pengujian ini dapat dilihat bahwa agen cerdas *Negascout* mengunjungi lebih sedikit node dalam pencarian langkah dibandingkan *Alpha Beta*. Dari segi waktu, *Alpha Beta* memerlukan lebih banyak waktu pencarian dibandingkan *Negascout*, ini berbanding lurus dengan jumlah eksplorasi node. Asumsi penulis dalam hasil yang didapatkan dari pengujian ini adalah dikarenakan pada algoritma *Negascout* dimungkinkan pemotongan pencarian node yang lebih baik dari *Alpha Beta*.

Dalam pengujian ini keadaan seri adalah keadaan dimana posisi bidak dan langkah yang dilakukan hanya berputar-putar dalam suatu pola saja tanpa adanya kemenangan. Pada dasarnya permainan *Nine Men's Morris* ini adalah permainan yang lebih besar kemungkinannya untuk mendapatkan hasil seri dan pemain pertama (putih) memiliki peluang lebih besar untuk menang.

Berdasarkan hasil pengujian didapatkan bahwa pada kedalaman 2 dan 4, agen cerdas *Alpha Beta* dapat menahan imbang agen cerdas *Negascout*, dan pada kedalaman 3 *Negascout* dapat memenangkan pertandingan. Perlu diingat bahwa fungsi evaluasi kedua agen cerdas sama, sehingga nilai langkah terbaik yang diambil adalah sama.

Asumsi yang didapatkan adalah pada kedalaman 2, masih merupakan kedalaman yang dangkal, sehingga langkah yang diambil oleh kedua agen cerdas masih sama dan menyebabkan permainan menjadi seri. Pada kedalaman 3, merupakan kedalaman yang efektif bagi agen cerdas *Negascout*, sehingga agen cerdas dapat memenangkan permainan. Pada kedalaman 4, permainan kembali seri dikarenakan lebih banyak adanya pemotongan cabang yang dilakukan oleh kedua agen cerdas sehingga langkah yang diambil masih terbilang sama. Hal ini dapat dilihat dari lama waktu yang dibutuhkan untuk mengambil langkah pada kedalaman 4 lebih sedikit dibandingkan lama waktu pada kedalaman 3 pada kasus tertentu.

4.3 Hasil Pengujian Negascout Melawan Negascout

Pengujian ini dilakukan dengan mengujikan antar agen cerdas *Negascout* dengan berbagai macam variasi kedalaman dari kedalaman 2 sampai dengan 4. Kedua agen cerdas *Negascout* tersebut dilabelkan dengan *Negascout 1* dan *Negascout 2*.

Hasil pengujian merupakan hasil pembulatan perhitungan rata-rata tiap langkah pada suatu pertandingan. Pada tiap langkah (*move*) sistem menghitung berapa jumlah node yang dikunjungi dan berapa waktu yang dibutuhkan untuk menentukan langkah terbaik (dalam satuan milidetik).

Berdasarkan hasil pengujian didapatkan bahwa dalam semua kedalaman dari 2 sampai 4 menghasilkan hasil pertandingan yang seri. Perlu diingat bahwa fungsi evaluasi kedua agen cerdas sama, sehingga nilai langkah terbaik yang diambil adalah sama.

Asumsi yang didapatkan adalah tingkat kedalaman dalam pengujian ini hanya sedikit mempengaruhi dalam pengambilan langkah terbaik jika menggunakan algoritma *Negascout* yang sama dengan fungsi evaluasi yang sama pula. Hasil ini juga didapatkan karena pada dasarnya permainan *Nine Men's Morris* kebanyakan memiliki hasil seri.

4.4 Hasil Pengujian Tingkat Kedalaman

Pengujian ini dilakukan dengan mengadakan pertandingan antara agen cerdas *Negascout* melawan pengguna dengan berbagai tingkat kedalaman. Pengujian ini dilakukan dengan tingkat kedalaman dari 2 sampai 4.

Hasil pengujian merupakan hasil pembulatan perhitungan rata-rata tiap langkah pada suatu pertandingan. Pada tiap langkah (*move*) sistem menghitung berapa jumlah node yang dikunjungi dan berapa waktu yang dibutuhkan untuk menentukan langkah terbaik (dalam satuan milidetik).

Berdasarkan hasil pengujian didapatkan bahwa dengan kedalaman 2, *user* masih dapat memenangkan pertandingan dengan mudah. Pada kedalaman 3 dan 4, *user* mulai mengalami kesulitan dalam menghadapi agen cerdas.

Tingkat kedalaman jelas mempengaruhi lama waktu yang diperlukan serta jumlah node yang dikunjungi. Asumsi yang didapatkan adalah pada kedalaman 3 dan 4 tidak terdapat perbedaan yang besar dalam langkah terbaik yang diambil, tetapi dengan kedalaman 4 membutuhkan waktu yang lebih lama dan mengunjungi node yang lebih besar dari pada kedalaman 3 sehingga kedalaman 3 lebih optimal dibandingkan kedalaman 4.

5. Kesimpulan

Berdasarkan hasil implementasi dan analisis sistem, maka diperoleh kesimpulan sebagai berikut:

- a. Sistem pencarian langkah terbaik dalam permainan *Nine Men's Morris* menggunakan algoritma *Negascout* dapat memotong lebih banyak node dibandingkan dengan algoritma *Alpha-Beta Pruning*.
- b. Sistem pencarian langkah terbaik dalam permainan *Nine Men's Morris* menggunakan algoritma *Negascout* memerlukan waktu yang lebih singkat dalam pencariannya dibandingkan dengan algoritma *Alpha-Beta Pruning*.
- c. Sistem pencarian langkah terbaik dalam permainan *Nine Men's Morris* menggunakan algoritma *Negascout* memiliki jumlah kemenangan paling besar jika diterapkan dengan kedalaman sebesar 3 melawan algoritma *Alpha-Beta Pruning*.

6. Saran

Saran yang diajukan oleh penulis untuk pengembangan sistem demi mencapai hasil yang lebih baik dan mendapatkan temuan-temuan yang baru adalah:

- a. Tampilan dan interaksi sistem masih sangat minim, sehingga masih diperlukannya pengembangan sistem terutama pada animasi dan tampilan antarmuka sistem.
- b. Mengeksplorasi lebih jauh lagi fungsi evaluasi agar agen cerdas dapat lebih cepat dan lebih tepat dalam menentukan langkah terbaik.

Daftar Pustaka

- Boyd, M., & Hirunthanakorn, C. (2012). Analyzing Nine Men's Morris For a Optimal Strategy. 1-7.
- Gasser, R. (1996). Solving Nine Men's Morris. *MSRI* , 101-113.
- Millington, I., & Funge, J. (2009). *Artificial Intelligence for Games Second Edition*. Burlington: Morgan Kaufmann.
- Petcu, S.-A., & Holban, S. (2008). Nine Men's Morris: Evaluation Functions. *International Conferncs on Development and Application Systems* , 89-92.
- Reinefeld, A. (1983). An Improvement to The Scout Tree Search Algorithm . *International Computer Chess Association* , 4-14.
- Russel, S. J., & Norvig, P. (2003). *Artificial Intelligece : A Modern Approach Second Edition*. New Jersey: Prentice Hall.