

# Penerapan Algoritma *Branch and Bound* pada *Board Game* Fanorona

Muhammad Ecky Rabani - 13510037  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13510037@std.stei.itb.ac.id

**Abstrak**—Fanorona adalah sebuah *board game* tradisional dari Madagascar. Permainan ini tidak banyak dikenali masyarakat karena umurnya yang sudah tua. Fanorona dapat dimainkan oleh dua orang pemain. Banyak strategi yang bisa diterapkan dalam permainan ini demi mencapai kemenangan. Dalam makalah ini dapat dilihat penerapan algoritma *branch and bound* sebagai salah satu strategi untuk memenangkan permainan Fanorona. Teknik *branch and bound* juga bisa diterapkan sebagai *artificial intelligence* dalam permainan Fanorona versi digital.

**Kata kunci**—Fanorona, *Branch and Bound*, *Breadth-First Search*, *Goal Node*, *Uniform Cost Search*, *Kondisi Menang*, *Kondisi Kalah*, *Paika Moves*.

## I. PENDAHULUAN

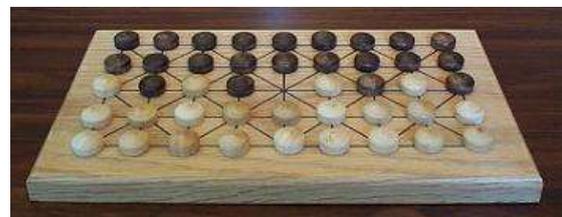
Permainan papan (*board game*) merupakan jenis permainan yang dimainkan di atas sebuah papan menggunakan sebuah objek, pion, atau potongan (*pieces*) yang dapat dipindah-pindahkan posisinya dari wilayah tertentu ke wilayah lain di atas permukaan papan sesuai dengan aturan tertentu. Seperti permainan pada umumnya, terdapat sebuah tujuan akhir (*goal*) yang harus dicapai oleh pemain. Biasanya, tujuan akhir sebuah permainan papan adalah memenangkan permainan dengan cara mengalahkan pemain lawan dari segi poin, skor, atau pencapaian kondisi tertentu yang menyebabkan kemenangan dan permainan berakhir. Permainan papan bervariasi mulai dari yang memiliki aturan paling sederhana seperti *Tic-Tac-Toe* sampai yang memiliki aturan paling rumit dan kompleksitas tinggi seperti *Monopoly*.



Gambar 1 Beragam Macam Jenis Permainan Papan

Contoh sebuah permainan papan yang kurang dikenal masyarakat adalah Fanorona. Fanorona merupakan permainan papan tradisional dari Madagascar. Umur dari permainan ini sudah sangat tua, dan permainan ini sudah memiliki banyak versi hasil dari modifikasi permainan Fanorona aslinya. Versi Fanorona yang paling umum adalah Fanorona dengan papan berukuran 5x9, nama lainnya adalah Fanoron-Tsivy. Ada beberapa versi legenda dari Madagascar yang populer mengenai permainan papan ini, seperti kisah raja Ralambo.

Fanorona hanya dapat dimainkan oleh dua orang pemain dimana tiap pemain akan mengontrol 22 bidak dengan warna yang berbeda untuk tiap pemain, biasanya berwarna hitam dan putih dimana putih mendapatkan giliran jalan pertama kali. Untuk memenangkan permainan ini dibutuhkan kemampuan dalam menyusun strategi dan taktik yang baik untuk diterapkan saat tiap pemain akan menggerakkan bidak per langkah-nya. Algoritma *branch and bound* yang dipelajari pada kuliah IF3051-Strategi Algoritma, dapat diterapkan pada setiap langkah saat bermain Fanorona. Algoritma *branch and bound* juga dapat dijadikan sebagai alternatif strategi yang dapat digunakan untuk memenangkan permainan papan ini.



Gambar 2 Tampilan Papan Permainan Fanorona

## II. DASAR TEORI

### A. Algoritma *Branch and Bound*

Algoritma *branch and bound* (B&B) merupakan metode pencarian solusi optimum di dalam sebuah ruang solusi secara sistematis. Ruang solusi tersebut diorganisasikan ke dalam sebuah pohon ruang status.

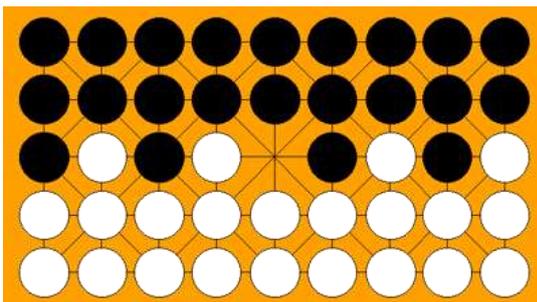
Pohon ruang status dalam algoritma *branch and bound* dibentuk dan dibangun secara dinamis berdasarkan skema *breadth-first search* (BFS). Algoritma *branch and bound* menggunakan teknik pemangkasan (*pruning*) dan tidak menerapkan teknik BFS murni dalam membangun pohon ruang statusnya. Karena hal ini algoritma *branch and bound* memiliki performansi yang lebih baik dalam pencarian solusi dibandingkan dengan algoritma BFS murni.

Untuk mempercepat pencarian ke sebuah simpul solusi, maka setiap simpul diberi sebuah nilai ongkos (*cost*). Simpul berikutnya yang akan diekspansi tidak lagi berdasarkan urutan pembangkitannya (sebagaimana pada BFS murni), tetapi simpul yang memiliki nilai ongkos optimum saat itu, baik nilai ongkos tersebut paling besar di antara simpul-simpul hidup lainnya, maupun kebalikannya (memiliki nilai ongkos paling kecil di antara simpul-simpul hidup lainnya). Dalam hal ini jika terdapat dua atau lebih simpul hidup yang memiliki nilai ongkos optimum, maka kedua simpul tersebut harus diekspansi.

Salah satu metode dalam algoritma *branch and bound* adalah modifikasi teknik *breadth-first search* (BFS) menjadi *uniform cost search* (UCS). Dalam teknik UCS, langkah yang dibutuhkan untuk mencapai simpul solusi (*goal node*) tidak lagi diperhitungkan, karena solusi dengan langkah paling sedikit belum tentu solusi paling optimum. Teknik UCS hanya memperhitungkan nilai ongkos dari tiap simpul hidup untuk menentukan simpul mana yang harus diekspansi. Untuk membantu dalam menentukan simpul hidup mana yang harus diekspansi, UCS menggunakan sebuah *queue* yang berisi semua simpul hidup terurut berdasarkan nilai ongkosnya.

### B. Permainan papan Fanorona

Fanorona dimainkan oleh dua orang pemain. Setiap pemain bebas mengontrol 22 bidaknya. Total seluruh bidak yang digunakan dalam permainan Fanorona berjumlah 44 bidak dengan dua warna yang berbeda, biasanya berwarna hitam dan putih. Pemain yang mengontrol bidak berwarna putih diperkenankan untuk jalan terlebih dahulu. Seluruh bidak diletakkan di papan permainan disetiap titik perpotongan empat atau lebih garis dengan format seperti pada gambar nomor 3.

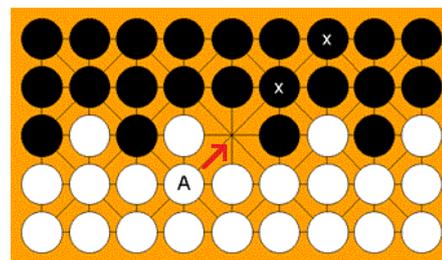


Gambar 3 Formasi Peletakkan Bidak Pada Awal Permainan Fanorona

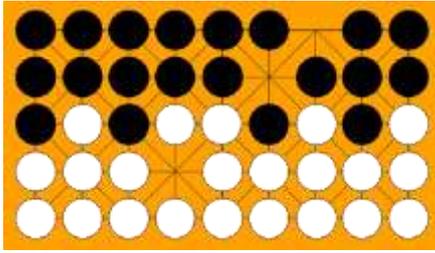
Untuk memenangkan pertandingan, pemain harus merebut (*capture*) seluruh bidak lawan seperti pada permainan papan *checkers*, atau memaksa lawan berada di posisi dimana lawan tidak bisa lagi menggerakkan bidaknya. Jika kedua pemain tidak ada yang bisa memenuhi kondisi menang tersebut, maka permainan berakhir dengan hasil seri.

Pada tiap giliran jalan, pemain diperbolehkan untuk memindahkan satu bidaknya ke wilayah lain dalam papan permainan yang kosong, bersebelahan dan merupakan titik perpotongan antara empat atau lebih garis. Tidak boleh ada dua bidak atau lebih yang bertumpuk di satu wilayah. Pergerakan bidak mengikuti salah satu garis yang berpotongan sebagai panutan arah perpindahan. Jika hanya terdapat garis horisontal dan vertikal yang berpotongan (*weak points*), maka bidak tersebut hanya bisa bergerak ke kiri, kanan, atas, dan bawah (empat arah). Jika terdapat garis vertikal, horisontal, dan diagonal yang berpotongan (*strong points*), maka bidak tersebut dapat bergerak secara diagonal ke kanan-atas, kanan-bawah, kiri-atas, kiri-bawah, ditambah empat gerakan seperti bidak pada *weak points* (delapan arah).

Terdapat dua jenis teknik perebutan bidak lawan. Pemain dapat merebut bidak lawan dengan cara menghampiri langsung bidak lawan (*capture by approach*) saat pemain mendapatkan giliran jalan. Semua bidak lawan yang berdekatan dengan bidak tersebut juga ikut direbut selama bidak tersebut berada di sepanjang garis yang menjadi panutan arah pergerakan bidak pemain dan diantaranya tidak ada wilayah kosong atau bidak pemain itu sendiri. Pemain juga dapat merebut bidak lawan dengan cara menjauh dari bidak lawan yang bersebelahan (*capture by withdrawal*) saat pemain mendapatkan giliran jalan. Sama seperti sebelumnya, semua bidak lawan yang berdekatan dengan bidak tersebut juga ikut direbut selama bidak tersebut berada di sepanjang garis yang menjadi panutan arah pergerakan bidak pemain, namun dengan arah yang berkebalikan, dan diantaranya tidak ada wilayah kosong atau bidak pemain itu sendiri.

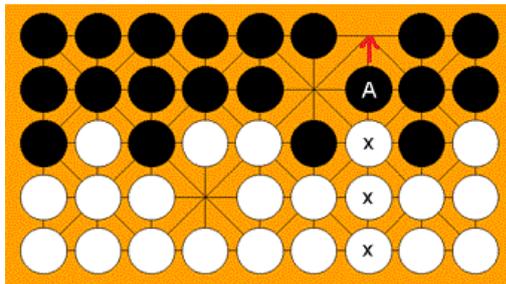


Gambar 4 Formasi Sebelum Pemain Menggerakkan Bidaknya (Putih)

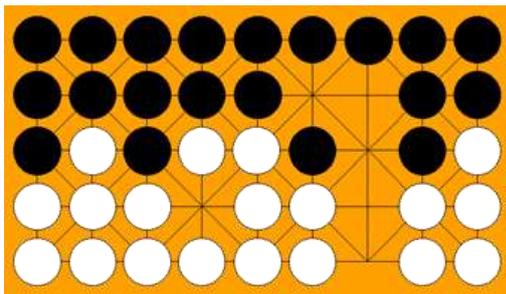


Gambar 5 Formasi Setelah Pemain Menggerakkan Bidaknya (Putih)

Pada gambar diatas dapat dilihat contoh teknik *capture by approach*. Misal pemain dengan bidak berwarna putih sedang mendapatkan giliran jalan. Pemain tersebut memindahkan bidak 'A' ke wilayah kosong di tengah dan bergerak secara diagonal. Pemain tersebut berhak mengambil bidak berwarna hitam 'x' sesuai dengan aturan *capture by approach*. Untuk contoh teknik *capture by withdrawal* dapat dilihat pada gambar nomor 6 dan gambar nomor 7.

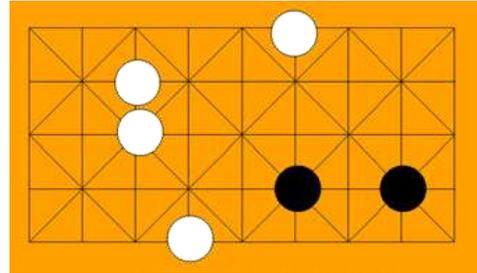


Gambar 6 Formasi Sebelum Pemain Menggerakkan Bidaknya (Hitam)



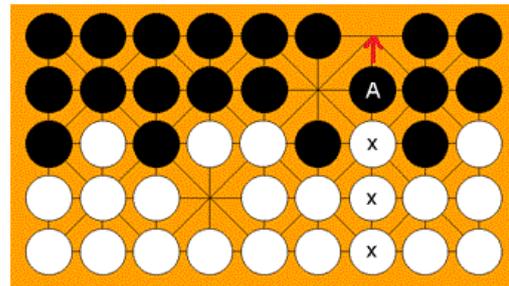
Gambar 7 Formasi Setelah Pemain Menggerakkan Bidaknya (Hitam)

Setiap pemain yang mendapatkan giliran jalan diharuskan merebut bidak lawannya. Jika pemain berada dalam posisi dimana pada suatu giliran pemain tidak dapat merebut bidak lawan, maka pemain tersebut diizinkan untuk menggerakkan bidak manapun miliknya ke wilayah kosong lain yang bersebelahan pada papan permainan. Kondisi seperti ini disebut gerak *paika* (*paika moves*).

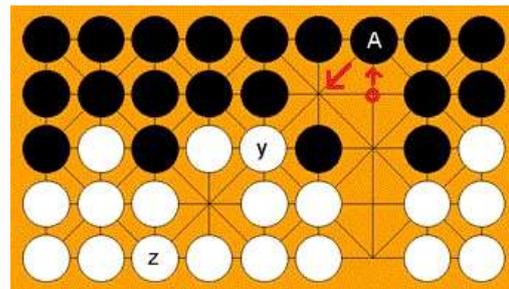


Gambar 8 Contoh Saat Terjadi Paika Moves

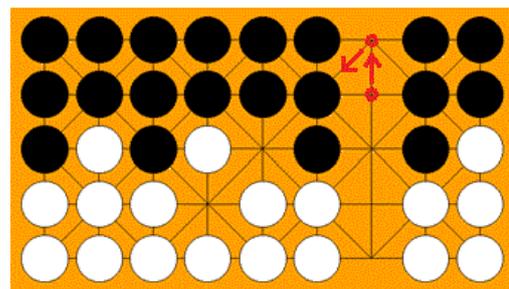
Pemain diizinkan untuk menggerakkan bidaknya lebih dari satu kali jika masih dimungkinkan bagi pemain untuk merebut bidak lawan lainnya selama arah pergerakannya berbeda dari sebelumnya dan tidak menempati wilayah sama untuk kedua kalinya dalam proses perebutan bidak berantai tersebut (*relay capturing*). Contoh teknik *relay capturing* dapat dilihat pada gambar nomor 9 sampai gambar nomor 11.



Gambar 9 Formasi Sebelum Pemain Menggerakkan Bidaknya (Hitam)



Gambar 10 Formasi Setelah Pemain Melakukan Gerakan Pertama (Hitam)

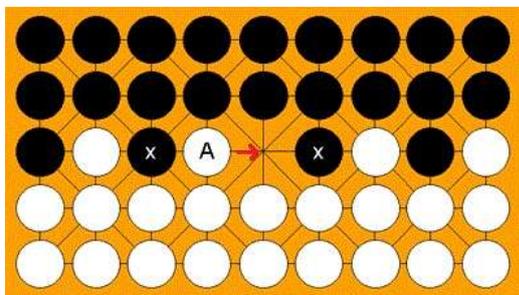


Gambar 11 Formasi Setelah Pemain Melakukan Gerakan Kedua (Hitam)

Pada serangkaian gambar diatas dapat dilihat contoh teknik *relay capturing*. Misal pemain dengan bidak berwarna hitam sedang mendapatkan giliran jalan. Pada gambar nomor 9, pemain tersebut akan memindahkan

bidak 'A' ke wilayah kosong di atas dan bergerak secara vertikal. Pemain tersebut berhak mengambil bidak berwarna putih 'x' sesuai dengan aturan *capture by withdrawal*. Pada gambar nomor 10 dapat dilihat bahwa ternyata, jika pemain tersebut memindahkan lagi bidak 'A' ke wilayah kosong di bawah dan bergerak secara diagonal, maka pemain tersebut dapat merebut bidak berwarna putih 'y' sesuai dengan aturan *capture by approach*. Bidak berwarna putih 'z' tidak dapat direbut karena terdapat wilayah kosong diantara bidak 'y' dan 'z', seperti yang disebutkan dalam aturan *capture by approach*. Karena hal tersebut, maka pemain tersebut mendapatkan giliran jalan lagi dan memindahkan bidak 'A' ke wilayah bawah tersebut sambil merebut bidak berwarna putih 'y' sesuai dengan aturan *relay capturing*. Pada beberapa versi lain permainan Fanorona, ada aturan yang menyebutkan bahwa jika ada pemain yang mendapatkan giliran jalan tambahan melalui *relay capturing*, pemain tersebut tidak wajib memindahkan bidak yang sedang aktif tersebut. Pemain bisa saja memilih untuk diam ditempat dan mempersilahkan lawan mendapatkan giliran jalannya, sehingga pergerakan bidak setelah melakukan satu kali gerakan bersifat opsional.

Ada beberapa kondisi spesial dalam Fanorona. Contoh kondisi spesial dapat dilihat pada gambar nomor 12.



Gambar 12 Contoh Kondisi Spesial Dalam Permainan Fanorona

Pada gambar diatas dapat dilihat contoh kondisi spesial dalam permainan Fanorona. Misal pemain dengan bidak berwarna putih sedang mendapatkan giliran jalan. Pemain tersebut akan memindahkan bidak 'A' ke wilayah kosong di sebelah kanan dan bergerak secara horisontal. Menurut aturan *capture by approach* dan *capture by withdrawal*, pemain tersebut diizinkan untuk merebut kedua bidak berwarna hitam 'x'. Namun, ada aturan dalam permainan Fanorona dimana jika pemain sedang berada dalam kondisi seperti ini, pemain tersebut hanya diizinkan mengambil salah satu bidak saja dan juga bidak lawan lain yang bersebelahan di sepanjang garis tersebut. Jika seperti pada gambar di atas, pemain tersebut harus memilih akan merebut bidak berwarna hitam 'x' yang mana, apakah yang letaknya di sebelah kanan bidak 'A' atau sebelah kiri bidak 'A'. Harus diingat bahwa teknik *relay capturing* tidak berlaku disini, sehingga pemain tersebut tidak dapat mengambil kedua bidak 'x' dalam dua kali gerakan atau giliran jalan. Hal ini karena saat proses perebutan bidak

berantai tersebut, pemain dilarang menempati wilayah yang sama dua kali sesuai dengan aturan *relay capturing*.

### III. ANALISIS MASALAH

Algoritma *branch and bound* dapat diterapkan pada permainan papan Fanorona untuk menentukan langkah pemain setiap mendapatkan giliran jalan. Jika pemain ingin merebut bidak lawan sebanyak-banyaknya setiap pemain tersebut mendapatkan giliran jalan, maka setiap kemungkinan langkah yang dapat dilakukan saat itu dapat dijadikan sebuah pohon ruang status permasalahan dengan akar (*root*) dari pohon tersebut adalah kondisi papan permainan saat tiba giliran pemain untuk jalan. Nilai ongkos dari *root* adalah 0 ( $C(0) = 0$ ), sedangkan nilai ongkos untuk simpul anak-anaknya dapat kita definisikan sebagai:

$$C(i) = f(i)$$

Dalam hal ini,

$C(i)$  = nilai ongkos untuk simpul ke-i

$f(i)$  = jumlah bidak lawan yang berhasil direbut

Pada tiap langkah algoritma *branch and bound*, dapat digunakan *queue* berisi semua simpul hidup saat ini yang sudah terurut berdasarkan nilai ongkos dari tiap simpul. Karena dalam permainan Fanorona pemain diharuskan merebut bidak lawan sebanyak-banyaknya untuk memenangkan pertandingan, maka *queue* tersebut akan terurut mengecil sehingga akan dilakukan ekspansi dari simpul hidup yang memiliki nilai ongkos terbesar setiap langkahnya. Jika telah sampai pada tahap dimana simpul hidup yang harus diekspansi ternyata berupa daun dari pohon ruang status, maka simpul tersebut merupakan simpul solusi (*goal node*) dari permasalahan ini dan pemain akan memindahkan bidaknya sesuai dengan langkah pada penelusuran solusi (menelusuri dari *goal node* ke *root*). Setelah lawan menyelesaikan gilirannya, pemain dapat melangkah lagi dengan kembali menerapkan algoritma *branch and bound* dengan membentuk pohon ruang status baru dengan *root*-nya adalah kondisi papan permainan saat itu.

Berikut adalah penjabaran dalam *pseudo-code* algoritma *branch and bound* untuk menentukan langkah dalam permainan papan Fanorona:

```

procedure PenentuanLangkah (input R : kondisi, output S
: langkah)
{
  Queue Q;
  pohon P;

  Q.Enqueue(R);
  P.Create(R);

  while (Next(Q.Dequeue) != NULL) do

```

```

{
  listsimpul[] L;
  L = P.Expand(Q.Dequeue); /* termasuk simpul
                           yang bernilai ongkos sama */
  Q.Enqueue(L);
  Q.SortDesc();
}

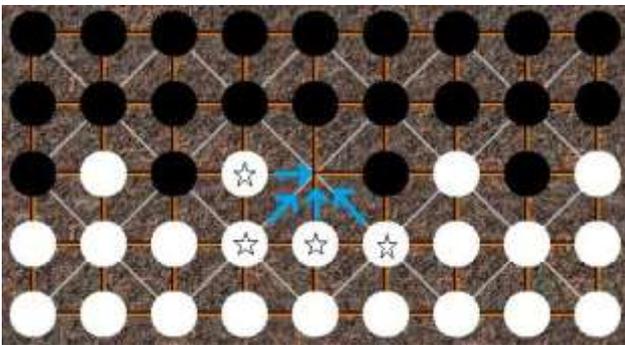
S = P.Telusur(R,Q.Dequeue);

return S;
}

```

IV. PENERAPAN ALGORITMA BRANCH AND BOUND PADA PERMAINAN FANORONA

Misalkan dimulai sesi permainan Fanorona baru dan penerapan algoritma *branch and bound* akan dilakukan pada pemain dengan bidak berwarna putih. Pemain tersebut mendapatkan giliran jalan pertama. Kondisi papan permainan saat ini akan dijadikan *root* dari pohon ruang status dengan nilai ongkos berjumlah 0 ( $C(0) = 0$ ).



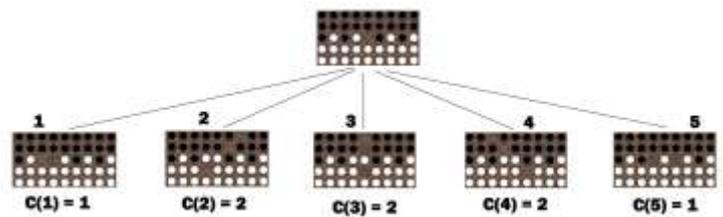
Gambar 13 Root Dari Pohon Ruang Status

Hasil ekspansi dari simpul *root* akan menghidupkan empat buah simpul anak baru. Dapat dilihat dari gambar nomor 13 di atas bahwa pemain dengan bidak berwarna putih mempunyai empat bidak pilihan yang bisa dipindahkan ke wilayah kosong pada giliran jalannya saat ini. Nilai ongkos untuk setiap simpul hidup baru tersebut adalah:

- $C(1) = 1$
- $C(2) = 2$
- $C(3) = 2$
- $C(4) = 2$

Untuk simpul ke-1, terdapat dua pilihan bidak berwarna hitam yang bisa direbut. Namun dalam hal ini, bidak manapun yang dipilih, jumlah maksimum bidak lawan yang dapat diambil tetap satu, karena tidak ada bidak lawan lain yang bersebelahan di sepanjang garis tersebut. Tapi tetap saja kondisi ini harus ditulis sebagai simpul baru dalam pohon ruang status agar semua kemungkinan terkomputasi dan kemungkinan mendapatkan nilai optimal lebih tinggi. Karena itu dibuat simpul anak baru yaitu  $C(5)$

yang memiliki nilai ongkos sejumlah satu. Berikut adalah gambar pohon ruang status sementara.



Gambar 14 Pohon Ruang Status Sementara

Untuk saat ini simpul hidup dalam *queue* adalah simpul ke-1, simpul ke-2, simpul ke-3, simpul ke-4, dan simpul ke-5. Simpul yang memiliki nilai ongkos terbesar adalah simpul ke-2, simpul ke-3, simpul ke-4 dengan nilai ongkos sebesar 2. Untuk selanjutnya, tiga simpul tersebut akan diekspansi lagi. Namun dari ketiga kondisi tersebut, tidak ada yang memungkinkan pemain untuk melakukan *relay capturing*. Biasanya algoritma *branch and bound* akan memilih simpul dengan nomor terkecil jika terdapat lebih dari satu *goal node*, hal ini tergantung dari struktur instruksi dalam algoritma *branch and bound* yang diterapkan dalam permainan Fanorona ini. Setelah lawan menyelesaikan giliran jalannya, akan dibuat sebuah pohon ruang status baru dengan simpul *root*-nya berupa kondisi papan permainan sekarang. Langkah ini dilakukan terus-menerus hingga salah satu pemain mencapai kondisi menang atau kedua pemain mencapai kondisi *paika moves*.

Untuk beberapa kasus lainnya, algoritma *branch and bound* ini akan menghasilkan solusi yang belum pasti optimum. Meskipun begitu, algoritma ini dapat membantu pemain menentukan langkah yang harus dilakukan agar dapat merebut bidak lawan sebanyak mungkin. Dalam pengembangannya mungkin algoritma *branch and bound* ini dapat diterapkan menjadi sebuah *artificial intelligence* untuk permainan papan Fanorona versi digital.

V. KESIMPULAN

Algoritma *branch and bound* dapat diterapkan pada permainan papan Fanorona untuk membantu pemain menentukan langkah yang optimal dimana pemain tersebut dapat merebut bidak lawan sebanyak mungkin ketika mendapatkan giliran jalannya. Hal tersebut sangat berpengaruh dalam permainan Fanorona karena pemain yang dinyatakan sebagai pemenang adalah pemain yang berhasil merebut semua bidak lawannya.

Kekurangan dari algoritma *branch and bound* adalah algoritma ini belum tentu menghasilkan solusi yang optimum. Algoritma ini juga tidak dapat diterapkan dalam kondisi-kondisi spesial yang ada di dalam permainan Fanorona seperti saat kondisi *paika moves* dimana tidak ada pemain yang bisa merebut bidak lawannya ketika mendapatkan giliran jalan. Butuh pengembangan lebih lanjut agar algoritma *branch and bound* ini dapat bekerja

lebih efektif dalam penerapannya pada permainan papan Fanorona.

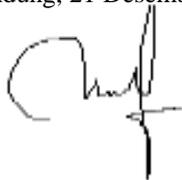
#### REFERENCES

- [1] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Bandung, 2009.
- [2] [http://rescomp.stanford.edu/~domingo2/stuff/fanorona\\_java.html](http://rescomp.stanford.edu/~domingo2/stuff/fanorona_java.html)
- [3] <http://www.oakgames.com/fanoronasmall.JPG>
- [4] <http://ilk.uvt.nl/icga/games/Fanorona/>
- [5] [http://gasy-fanorona.sourceforge.net/docs/fanorona\\_rules.html](http://gasy-fanorona.sourceforge.net/docs/fanorona_rules.html)
- [6] <http://www.ccg.com/games/index.htm>
- [7] <http://www.academic.marist.edu/~jzbv/algorithms/Branch%20and%20Bound.htm>

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 21 Desember 2012



Muhammad Ecky Rabani - 13510037