

# Aplikasi struktur data *The Minimax Game Tree* pada permainan catur

Abdul Gapur

*Departemen Teknik Informatika.*

*Institut Teknologi Bandung*

*Jl. Ganesha 10, Bandung*

Email : [if15011@students.if.itb.ac.id](mailto:if15011@students.if.itb.ac.id)

## Abstrak

Permasalahan yang timbul selalu diharapkan mempunyai solusi yang bagus. Walaupun sebenarnya algoritma selalu berkembang untuk suatu permasalahan-permasalahan yang berbeda. Perkembangan algoritma sesuai dengan kebutuhannya pada dunia inteligensi buatan. Permainan yang berbasis pada giliran pemain. Salah satu algoritma yang cukup mangkus pada saat ini adalah Branch and Bound yang diimplementasikan pada struktur data *The Minimax Game Tree*. Dengan konsep utama terletak pada semua kemungkinan yang bisa ditelusuri pada permainan. Dalam hal ini, permainan catur dapat menggunakan algoritma ini karena permainan dapat diimplementasikan dalam sebuah pohon. Pohon yang mempunyai cabang yang terdiri dari simpul-simpul yang menyatakan status-status yang nantinya digunakan untuk menentukan langkah dalam pencarian tujuan permainan.

Kata kunci : *The Minimax Game Tree, searching, current state.*

## Pendahuluan

Alasan struktur data ini dinamakan *The Minimax Game Tree* adalah karena algoritma sederhana yang ada dibelakang struktur ini. Misalnya kita memberi suatu tanda pada permainan *Tic-Tac-Toe*. Jika X yang memenangkan permainan maka situasi permainan di berikan dengan angka 1. Jika O yang menang, maka permainan mempunyai nilai -1. Sekarang, X akan mencoba untuk memaksimalkan skor yang telah diberikan dan O akan berusaha meminimalkan skor yang diberikan ke padanya. Jadi, salah satu peneliti pertama algoritma ini menamainya *The Minimax Game Tree*. Oleh karena itu, secara keseluruhan struktur datanya dinamakan *The Minimax Game Tree*.

*The Minimax Game Tree* tidak hanya digunakan pada permainan *Tic-Tac-Toe*. Struktur data ini juga digunakan pada permainan catur. Karena pada permainan catur dibutuhkan algoritma yang bisa menentukan jalan yang terbaik saat itu. Pada permainan yang lebih rumit, kita tidak pernah bisa melihat secara keseluruhan isi dari situasi permainan yang telah diimplementasikan sebagai

pohon bercabang. Maka, komputer hanya memperhatikan semua simpul yang telah ada dan mencari simpul yang baik, dalam hal ini simpul yang mempunyai langkah terbaik untuk mencapai tujuan permainan. Kemudian komputer mencoba untuk membuang atau memangkas simpul yang tidak mengarah ke solusi pada saat itu.

Strategi di belakang Minimax algoritma adalah komputer berasumsi bahwa kedua pemain akan main sejauh kemampuan mereka. Maka, jika lawan mempunyai pilihan suatu gerak tidak baik atau suatu yang baik gerak, komputer akan mempunyai lawan memilih yang baik gerak. sungguh mencoba untuk memilih gerak yang mengakibatkan nilai titik yang paling rendah. Konsep relatif nyata dan sederhana ini menjadi rahasia di belakang minimax pohon. Jika komputer di program untuk mencari maksimal, maka rangkaian atau gerakan yang terbaik akan selalu ditemukan yang mengakibatkan nilai titik yang paling tinggi. Demikian juga dengan minimal, maka rangkaian atau gerakan yang terbaik akan selalu ditemukan yang mengakibatkan nilai titik yang paling rendah.

## Tujuan penulisan makalah

Adapun tujuan dari penulisan makalah ini adalah untuk memperkenalkan algoritma yang dipakai untuk permainan catur. Pada makalah ini dikenalkan cara kerja algoritma *The Minimax Game Tree* kepada para pembaca. Sehingga pembaca dapat mengkaji lebih lanjut tentang peranan algoritma ini pada permainan tertentu, kelemahan dan kelebihan. Makalah ini juga dituliskan untuk memenuhi tugas mata kuliah IF2251 Strategi Algoritmik pada semester genap tahun ajaran 2006/2007 di Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

## Ruang Lingkup Permasalahan

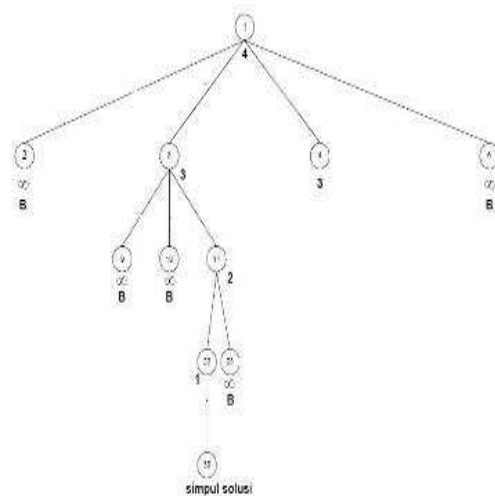
Ruang lingkup dari permasalahan adalah penggunaan algoritma *The Minimax Game Tree* untuk permainan yang berbasis pada giliran permainan. Dalam hal ini permainan yang diangkat adalah permainan catur. Pada permainan ini dimainkan oleh dua aktor secara bergantian. Misalnya pemain dengan pemain atau pemain dengan komputer.

## Pembahasan

Salah satu struktur data yang paling utama dalam pembuatan permainan adalah minimax pohon. Hampir di setiap *board games* algoritma ini digunakan. Salah algoritma yang dipakai pada struktur data ini adalah *branch and bound*. *The Minimax Game Tree* digunakan untuk memainkan permainan yang biasanya terlibat dua pemain sehingga permainan di jalankan secara bergantian. Seperti pada permainan *Tic-Tac-Toe* (tebak-tabakan). Secara fisik pohon ini terdiri dari simpul dari semua kemungkinan gerakan. Jadi pohon ini tidak sama dengan pohon yang biasa kita kenal seperti pohon biner, pohon 3, atau pohon 4. *The Minimax Game Tree* mempunyai jumlah cabang yang sangat bervariasi, jumlah cabang ditentukan oleh situasi permainan.

Tingkatan dari pohon tersebut tergantung dari tipe permainan yang dikembangkan. Biasanya dalam permainan mengarah pada min atau max. dalam hal ini min atau max adalah tujuan dari permainan misalnya pada labirin tujuannya adalah untuk mencari jalan yang terpendek keluar dari permainan.

*The Minimax Game Tree* penuh, komputer bisa menentukan semua kemungkinan langkah yang terbaik. Tetapi dengan konsekuensi bahwa dengan gerak yang sedikit saja pohon akan meledak dengan cepat. Hal ini tentu tidak efisien untuk permainan yang berskala besar seperti catur. Komputer terpaksa manaksir siapa yang menang dan yang kalah dengan melihat dari keseluruhan pohon yang sangat besar tersebut. Untuk mengatasi kendala tersebut maka programmer memberikan tambahan algoritma seperti *Alpha-Beta*, *negascout*, dan *MTD(f)*. Sehingga diharapkan berkurangnya cabang dari pohon tersebut.



gambar 1. Pohon yang di ekspansi pada cabang tertentu [1]

Tidak semua permainan yang bersifat searching dapat menggunakan algoritma ini. Seperti pada permainan poker(kartu). Komputer akan sangat bekerja keras untuk menentukan apa yang akan dilakukan lawan karena komputer tidak dapat "melihat" secara keseluruhan situasi permainan. Maka, *The Minimax Game Tree* akan bekerja dengan baik untuk permainan *checkers*, *Othello*, *chess*, dan *go*. Karena informasi pada permainan-permainan tersebut lengkap. Komputer bisa tetap "memantau" kondisi permainan sewaktu permainan sedang berlangsung.

Algoritma yang di pakai pada kasus ini adalah algoritma pohon. Tepatnya algoritma BFS. Sebuah pohon permainan terdiri dari simpul-simpul. Setiap simpul mempunyai status yang berbeda yang bisa dikunjungi. Jadi, dari satu simpul kita bisa mengunjungi simpul-simpul berbeda lainnya.

Pada permainan catur, posisi pertama dimisalkan sebagai simpul pertama pada pohon, dari simpul tersebut kita bisa mengunjungi 20 simpul berbeda lainnya. Yaitu pada permulaan susunan catur, maka yang bisa dijalankan hanya delapan pawn yang bisa dijalankan satu langkah kedepan atau dua langkah kedepan, sedangkan knight bisa berjalan ke samping kanan dan samping kiri, jadi,  $8 \times 2 + 2 \times 2 = 20$ .



gambar 2. gambar papan catur

Setiap simpul mempunyai nilai heuristik, yaitu nilai yang ditentukan untuk bergerak dari simpul yang satu kesimpul berbeda lainnya. Dalam permainan ini nilai heuristik adalah jauhnya posisi satu simpul ke simpul tujuan misalnya, untuk mencapai raja atau sekakmat (*checkmate*). Jadi dengan nilai ini kita bisa menentukan langkah terbaik untuk menjalankan 16 simpul yang ada.

Untuk meningkatkan efektifitas dari nilai heuristik kita tidak hanya memperhatikan jumlah langkah terpendek untuk sampai ke raja, tetapi kita juga harus memperhatikan ada yang bisa berjalan jauh lebih dari satu langkah misalnya queen, benteng, rencong. Untuk permainan catur (permainan yang berbasis pada giliran pemain), nilai heuristik tidak bisa didasarkan pada kedekatan *current state* ke *goal state*.

Pada permainan ini kita dapat membuat perkiraan yang mendekati karena keseluruhan permainan (semua cabang pohon dapat dilihat). Semua status pada simpul akan berubah saat pemain menjalankan salah satu anak catur.

Jadi pada permainan catur ini, status yang sudah ditandai pada sebuah simpul tidak akan berubah oleh gerakan lawan. Karena pada permainan catur pemain tidak dapat mengeser status dari lawannya kecuali kita sendiri yang mengubah jalannya salah satu anak catur tersebut. Gerakan tersebut akan membuat status baru bisa berubah minimal status dari anak catur yang baru dijalankan karena langkah ke status tujuan dalam hal ini untuk sekakmat (*checkmate*).

Jadi algoritma ini akan terus menerus memperbarui semua simpul yang telah di tentukan statusnya. Karena dengan satu gerakan dari anak catur makan kemungkinan dari status simpul juga akan berubah, dengan memperhatikan kedekatannya dengan raja atau simpul tujuan.

### Kelebihan Algoritma The Minimax Game Tree.

Adapun kelebihan algoritma ini adalah mampu memberikan pendekatan solusi-solusi yang nantinya diharapkan dapat menjadi solusi yang diharapkan sebagai solusi yang baik dari permasalahan.

Dengan adanya penambahan algoritma pada pencarian nilai heuristik yang dibuat untuk menentukan status pada setiap simpul pohon, maka akan sangat membantu dalam mengurangi kemungkinan cabang yang ada. Komputer akan mengembangkan simpul yang kira-kira mengarah pada solusi persoalan. Dengan otomatis algoritma ini akan lebih efisien dalam menentukan solusi terbaik.

Sangat baik digunakan untuk permainan yang menggunakan dua pemain yang dapat memainkan permainan secara bergantian.

### Kelemahan Algoritma The Minimax Game Tree.

Dengan algoritma ini kita tidak dapat menentukan secara pasti apakah permainan akan dimenangkan oleh pemain satu atau pemain yang kedua. Karena dengan aplikasi algoritma ini hanya menaksir nilai terbaik dari setiap simpul yang ada pada saat itu. Sedangkan simpul tersebut dapat berubah status dengan gerakan anak catur.

Untuk permainan yang melawan komputer. Ada sedikit kekurangan dari algoritma ini yaitu konsistensinya dalam mencari solusi dari setiap masalah. Untuk grandmaster catur yang telah terbiasa bermain dengan algoritma ini akan menemukan sedikit celah akibat konsistensi algoritma dalam menemukan solusi.

## Kesimpulan

Algoritma ini dapat dimanfaatkan sebagai otak dari permainan dalam inteligensi buatan. Dengan adanya algoritma ini. Salah satu dari jenis permasalahan yang melibatkan pencarian (*searching*) akan lebih mudah dan lebih efisien dibandingkan beberapa algoritma yang sejenis yang menggunakan pohon seperti DFS atau BFS.

Masih terdapat kekurangan pada algoritma *The Minimax Game Tree* ini. Seperti yang terdapat pada pembahasan yang telah dipaparkan diatas. Tetapi walaupun pada saat ini The Minimax Game Tree merupakan salah satu algoritma yang mangkus untuk menyelesaikan permainan yang berbasis

giliran pada pemain pada waktu permainan berlangsung

## Daftar Pustaka

- [1] Munir, Rinaldi.2007. "*Strategi Algoritmik*".  
Departemen Teknik Informatika, Institut  
Teknologi Bandung.
- [2]<http://www.aihorizon.com/essays/basiccs/trees/minimax.htm>  
Diakses tanggal 22 Mei 2007
- [2]<http://www.aihorizon.com/essays/basiccs/trees/minimax3.htm>  
Diakses tanggal 22 Mei 2007
- [3]<http://ai-depot.com/Features/>  
Diakses tanggal 22 Mei 2007