

METODE IMPLEMENTASI POHON N-ARY DALAM ARTIFICIAL INTELLIGENCE GAME STUDI KASUS : MINIMAX PADA TIC TAC TOE

Andhika Hendra Estrada – NIM : 13505118

Program Studi Teknik Informatika, Institut Teknologi Bandung

Jl. Ganesha 10, Bandung

E-mail : if15118@students.if.itb.ac.id

Abstrak

Game industry adalah salah satu entertainment industry yang banyak sekali melibatkan peran scientist di bidang matematika dan informatika. Salah satu penerapannya adalah dalam pembuatan *Artificial Intelligence*. Berbicara tentang *Artificial Intelligence* atau kecerdasan buatan, salah satu teknologi komputer dan mesin yang terus berkembang ini merupakan salah satu bagian dari ilmu informatika yang mempunyai banyak sekali jenis algoritma.

Untuk pembentukan *Artificial Intelligence* pada game ternyata digunakan pula pohon n-ary untuk suatu struktur. Implementasi pohon (tree) ini biasa disebut game tree. Berdasarkan game tree inilah sebuah game disusun algoritma kecerdasan buaatannya. Artificial intelligence yang disematkan dalam sebuah game yang membentuk analisis game tree biasanya merepresentasikan kondisi atau posisi permainan dari game sebagai suatu node, dan merepresentasikan langkah yang mungkin dilakukan sebagai sisi berarah yang menghubungkan node kondisi tersebut ke anak (child) sebagaimana representasi suatu pohon (tree). Namun, biasanya representasi langsung tersebut mempunyai kelemahan, yaitu representasi data pohon akan menjadi sangat lebar dan banyak. Mungkin bagi sebuah mesin komputer mampu melakukan kalkulasi sebanyak apapun masalah, namun game tree yang lebar dan besar memberikan beberapa masalah antara lain konsumsi proses memori, kapasitas penyimpanan yang cukup besar dan kinerja yang kurang pada console game berspesifikasi rendah.

Karena itu dibentuklah beberapa algoritma dan penyederhanaan bagi sebuah game tree. Pada salah satu contoh game klasik, yaitu tic tac toe, penyederhanaan dapat dilakukan dengan berbagai metode. Salah satu diantaranya adalah minimax. Metode ini berhasil diterapkan dan memberikan nilai reduksi yang cukup signifikan. Dan tidak hanya bisa digunakan secara monoton, minimax juga bisa digunakan untuk game-game yang lebih rumit seperti catur,. Tentunya dengan algoritma dan representasi berbeda.

Minimax yang merupakan salah satu metode penerapan (implementasi) pohon n-ary pada suatu game, menandakan bahwa implementasi struktur (pohon khususnya) sangatlah diperlukan pada pembuatan dan penerapan *Artificial Intelligence*, dan tidak menutup kemungkinan ilmu dan metode baru yang lebih canggih akan ditemukan di masa depan.

Kata kunci: *tree, game, Artificial Intelligence, minimax, tic tac toe, alghorithm, game tree, game industry.*

1. Pendahuluan

Seiring dengan kemajuan globalisasi, nuansa kompetitif makin kental dalam keseharian manusia. Seiring dengan itu, kecenderungan kegiatan didominasi oleh kegiatan-kegiatan yang lebih banyak mengkonsumsi stamina otak. Hal itu berbeda dengan kecenderungan kehidupan

manusia zaman dahulu (misalnya zaman prasejarah) yang masih di dominasi oleh kegiatan yang didominasi kinerja otot. Terkurusnya stamina otak yang dirasakan sebagian besar manusia zaman modern, tentu saja tidak cukup dipulihkan dengan istirahat fisik

saja. Karena itu kebutuhan entertainment sangatlah vital saat ini.

Seiring dengan majunya dunia entertainment, salah satu area entertainment yang cukup banyak melibatkan scientist dan artist adalah gaming industry. Dulunya game merupakan salah satu aspek entertainment yang minor, hanya sebagai selingan atau hiburan saja. Dan dianggap tidak menghasilkan sesuatu. Bahkan terkadang jika terdapat orang yang teramat sangat menggandrungi dunia game hal itu dianggap sesuatu yang tidak normal. Namun hal itu sedikit demi sedikit berubah. Ditandai dengan munculnya berbagai console yang cukup bervariasi menunjukkan bahwa dunia game tidak sedikitpun "mati", namun sedang berkembang dengan hebatnya. Game Watch kecil saku yang dulu pernah kita miliki, berkembang menjadi varian handphone dengan console game terintegrasi di dalamnya, lengkap dengan beberapa variasi kaset game yang kompatibel dengan handheld tersebut. Game computer klasik yang dulu pernah kita mainkan, kini beranjak menjadi sejumlah game online yang menghubungkan jutaan orang dari berbagai belahan dunia yang berbeda. Game konsol yang dulu bermesin kecil dan hanya mampu menampilkan kinerja gambar dua dimensi serta harus dimainkan bersama televisi di rumah, jauh berbeda dengan keberadaan sejumlah game console portable dengan spesifikasi sangat tinggi, yang memiliki layar sentuh. Hebatnya perkembangan dunia game saat ini tentu saja tidak berhenti di sini. Mungkin beberapa dekade kedepan perkembangan game sudah jauh di luar bayangan kita saat ini.

Jika dipandang dari segi konsumen game merupakan salah satu hal yang bisa jadi amat menarik, sampai-sampai bisa membuat seseorang "kecanduan". Namun tidak melulu hal negatif yang disuguhkan oleh gaming entertainment. Salah satu elemen mayor dunia entertainment modern ini menyuguhkan sejumlah hal positif yang bahkan sulit dicari di area entertainment lain.

Yang pertama adalah sifatnya yang bisa dinikmati oleh berbagai kalangan. Seorang balita (bayi di bawah lima tahun) yang belum bisa membaca dan bahkan belum bisa berbicara mungkin mampu berinteraksi secara baik dengan dunia game (permainan) ini. Bahkan bisa di katakan bahwa hal ini merupakan salah satu interaksi yang cocok untuk merangsang

pertumbuhannya baik secara fisik, mental, maupun persiapan akademis, karena definisi permainan atau game tidak melulu hanya duduk termangu berdiam diri di depan konsol permainan. Game juga berarti latihan yang dibentuk sebagai permainan interaksi nyata dengan orang-orang yang dekat dengan bayi tersebut (misalnya kasih sayang, perhatian, dan perawatan yang diberikan oleh ayah dan ibunya). Dan bukan tidak mungkin game console masa depan memfasilitasi hal ini. Tidak hanya kalangan di bawah umur, orang-orang tua pun terkadang dengan asyiknya terbenam dalam permainan-permainan klasik seperti tetris. Dan bukan tidak mungkin perkembangan dunia game masa depan menjurus pada game-game yang menitik beratkan pada penggunaan untuk usia lanjut, karena hal ini belum cukup di kembangkan untuk saat ini.

Kedua, terlibatnya otak dalam proses permainan game yang menjadikan game entertainment salah satu sarana potensial untuk mengembangkan dan melatih kemampuan, kreativitas, konsentrasi dan daya tahan otak manusia. Lebih dari itu jika hal ini dilakukan secara tepat sejak usia dini (sebagaimana hal positif pertama di atas), memungkinkan seseorang untuk mempunyai kemampuan otak yang lebih ketika beranjak dewasa, dan membuat orang tersebut sedikit lebih mudah memahami dan memasuki dunia pembelajaran akademis serta bersaing dan berinovasi di dunia kerja. Namun tentunya untuk hal ini diperlukan game yang mempunyai muatan positif dan mampu membangun karakter sebuah manusia. Karena itu kita harus selektif dalam memilih game apa yang boleh kita mainkan. Perkembangan dunia game yang sangat dahsyat dan jenis game yang bermacam-macam, mendorong terciptanya beberapa game yang "tidak membangun" dan memberikan lebih banyak dampak negatif terhadap perkembangan mental manusia.

Beberapa hal positif dari perkembangan game di atas dan ladang pekerjaan yang cukup menjanjikan di dunia gaming industry, menjadikan berbagai ilmu dan metode dalam produksi game entertainment merupakan salah satu ilmu yang patut dipelajari dan dikembangkan.

2. Artificial Intelligence

Meningkatnya spesifikasi, jenis dan teknologi yang mengiringi dunia game, menyebabkan terjadinya peningkatan selera masyarakat pada

suatu game. Mungkin jika dulu permainan versus (seorang pemain lawan pemain yang lain) masih cukup menarik untuk langsung diterapkan dan diproduksi dalam skala industri. Saat ini, harus diciptakan keberadaan *Artificial Intelligence* (kecerdasan buatan) demi selera game yang terus meningkat tersebut. Game yang monoton, gampang ditebak, dan flow cerita yang datar dan tidak berubah sudah tidak boleh ada di industri game saat ini.

Dari sisi konsumen tentu saja artificial intelligence menjadi aspek penting. Namun dari sisi produsen yang terjadi adalah sebaliknya. Mesin dan computer adalah benda mati yang tidak mempunyai kecerdasan dan kreativitas. Padahal console game dituntut untuk cerdas dan mampu melakukan tindakan-tindakan yang kreatif dan penuh pertimbangan. Karena itu dibentuklah *Artificial Intelligence*.

Kecerdasan Buatan (*Artificial Intelligence*) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (games), logika fuzzy, jaringan syaraf tiruan dan robotika.

Banyak hal yang kelihatannya sulit untuk kecerdasan manusia, tetapi untuk Informatika relatif tidak bermasalah. Seperti contoh: mentransformasikan persamaan, menyelesaikan persamaan integral, membuat permainan catur atau Backgammon. Di sisi lain, hal yang bagi manusia kelihatannya menuntut sedikit kecerdasan, sampai sekarang masih sulit untuk direalisasikan dalam Informatika. Seperti contoh: Pengenalan Obyek/Muka, bermain Sepakbola.

Walaupun AI memiliki konotasi fiksi ilmiah yang kuat, AI membentuk cabang yang sangat penting pada ilmu komputer, berhubungan dengan perilaku, pembelajaran dan adaptasi yang cerdas dalam sebuah mesin. Penelitian dalam AI menyangkut pembuatan mesin untuk mengotomatiskan tugas-tugas yang membutuhkan perilaku cerdas. Termasuk contohnya adalah pengendalian, perencanaan dan penjadwalan, kemampuan untuk menjawab diagnosa dan pertanyaan pelanggan, serta pengenalan tulisan tangan, suara dan wajah. Hal-

hal seperti itu telah menjadi disiplin ilmu tersendiri, yang memusatkan perhatian pada penyediaan solusi masalah kehidupan yang nyata. Sistem AI sekarang ini sering digunakan dalam bidang ekonomi, obat-obatan, teknik dan militer, seperti yang telah dibangun dalam beberapa aplikasi perangkat lunak komputer rumah dan video game.

'Kecerdasan buatan' ini bukan hanya ingin mengerti apa itu sistem kecerdasan, tapi juga mengkonstruksinya.

Tidak ada definisi yang memuaskan untuk 'kecerdasan':

1. kecerdasan: kemampuan untuk memperoleh pengetahuan dan menggunakannya
2. atau kecerdasan yaitu apa yang diukur oleh sebuah 'Test Kecerdasan'

Secara garis besar, AI terbagi ke dalam dua faham pemikiran yaitu AI Konvensional dan Kecerdasan Komputasional (CI, Computational Intelligence). AI konvensional kebanyakan melibatkan metoda-metoda yang sekarang diklasifikasikan sebagai pembelajaran mesin, yang ditandai dengan formalisme dan analisis statistik. Dikenal juga sebagai AI simbolis, AI logis, AI murni dan AI cara lama (GOF AI, Good Old Fashioned *Artificial Intelligence*). Metoda-metodanya meliputi:

1. Sistem pakar: menerapkan kapabilitas pertimbangan untuk mencapai kesimpulan. Sebuah sistem pakar dapat memproses sejumlah besar informasi yang diketahui dan menyediakan kesimpulan-kesimpulan berdasarkan pada informasi-informasi tersebut.
2. Petimbangan berdasar kasus
3. Jaringan Bayesian
4. AI berdasar tingkah laku: metoda modular pada pembentukan sistem AI secara manual

Kecerdasan komputasional melibatkan pengembangan atau pembelajaran iteratif (misalnya penalaan parameter seperti dalam sistem koneksionis. Pembelajaran ini berdasarkan pada data empiris dan diasosiasikan dengan AI non-simbolis, AI yang tak teratur dan perhitungan lunak. Metoda-metoda pokoknya meliputi:

1. Jaringan Syaraf: sistem dengan kemampuan pengenalan pola yang sangat kuat

2. Sistem Fuzzy: teknik-teknik untuk pertimbangan di bawah ketidakpastian, telah digunakan secara meluas dalam industri modern dan sistem kendali produk konsumen.

3. Komputasi Evolusioner: menerapkan konsep-konsep yang terinspirasi secara biologis seperti populasi, mutasi dan “survival of the fittest” untuk menghasilkan pemecahan masalah yang lebih baik.

Metoda-metoda ini terutama dibagi menjadi algoritma evolusioner (misalnya algoritma genetik) dan kecerdasan berkelompok (misalnya algoritma semut)

Dengan sistem cerdas hibrid, percobaan-percobaan dibuat untuk menggabungkan kedua kelompok ini. Aturan inferensi pakar dapat dibangkitkan melalui jaringan syaraf atau aturan produksi dari pembelajaran statistik seperti dalam ACT-R. Sebuah pendekatan baru yang menjanjikan disebutkan bahwa penguatan kecerdasan mencoba untuk mencapai kecerdasan buatan dalam proses pengembangan evolusioner sebagai efek samping dari penguatan kecerdasan manusia melalui teknologi.

Penemuan-penemuan baru tentang *Artificial Intelligence* terus berlanjut hingga saat ini, termasuk di dunia game.

3. Pohon (tree)

Definisi struktur data dari pohon:

Sebuah POHON adalah himpunan terbatas tidak kosong, dengan elemen yang dibedakan sebagai berikut :

- sebuah elemen dibedakan dari yang lain, yang disebut sebagai AKAR dari pohon
- elemen yang lain (jika masih ada) dibagi-bagi menjadi beberapa sub himpunan yang disjoint, dan masing-masing sub himpunan tersebut adalah POHON yang disebut sebagai SUB POHON dari POHON yang dimaksud.

Definisi pohon dari sudut pandang diskrit:

- pohon adalah graf terhubung berarah yang tidak memiliki sirkuit.

Struktur pohon adalah struktur yang penting dalam bidang informatika, yang memungkinkan kita untuk :

- mengorganisasi informasi berdasarkan suatu

struktur logik

- memungkinkan cara akses yang khusus terhadap suatu elemen

Contoh persoalan yang tepat untuk direpresentasi sebagai pohon:

- pohon keputusan,
- pohon keluarga dan klasifikasi dalam botani,
- pohon sintaks dan pohon ekspresi aritmatika

Contohnya, jika sebuah buku dipandang sebagai pohon. Judul buku adalah AKAR. Buku dibagi menjadi bab-bab. Masing-masing bab adalah sub pohon yang juga mengandung JUDUL sebagai AKAR dari bab tersebut. Bab dibagi menjadi sub bab yang juga

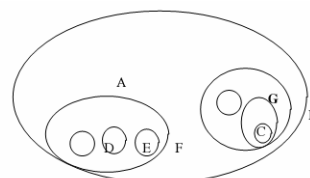
diberi judul. Sub bab adalah pohon dengan judul sub bab sebagai akar. Daftar Isi buku dengan penulisan yang di-identasi mencerminkan struktur pohon dari buku tersebut.

Dalam struktur data informatika terdapat pula istilah hutan. Definisinya adalah sequence (list) dari pohon). Jika kita mempunyai sebuah hutan, maka kita dapat menambahkan sebuah akar fiktif pada hutan tersebut dan hutan tersebut menjadi list dari sub pohon. Demikian pula sebaliknya: jika diberikan sebuah pohon dan kita membuang akarnya, maka akan didapatkan sebuah hutan.

Suffiks (akhiran) n-ary menunjukkan bahwa sub pohon bervariasi semua elemen dari pohon adalah akar dari sub pohon, yang sekaligus menunjukkan pohon tsb pada definisi di atas, tidak ada urutan sub pohon, namun jika logika dari persoalan mengharuskan suatu strukturasi seperti halnya pada buku, maka dikatakan bahwa pohon berarah

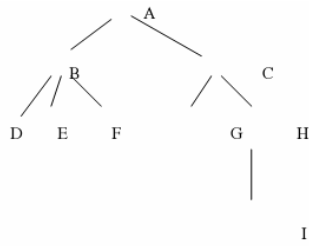
Untuk representasi yang lebih jelas, sebuah pohon dapat digambarkan. Berikut ini contoh representasi berbeda untuk pohon yang sama.

a) Himpunan



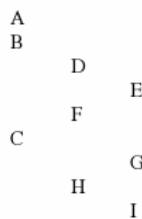
Gambar 1 Representasi Pohon menggunakan Himpunan

b) Graph



Gambar 2 Representasi Pohon menggunakan Graph

c) Indentasi



Gambar 3 Representasi Pohon menggunakan Indentasi

d) Bentuk linier :

Prefix : A (B (D , E , F) , C (G , H (I)))
 Posfix : ((D , E , F) B , (G , (I) H) C))

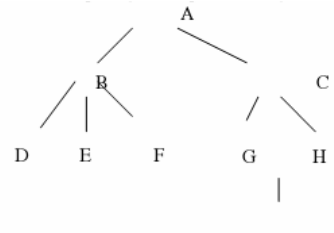
Dalam pohon atau hutan, kita juga menyebut bagian dari pohon dengan istilah-istilah tertentu.

- *SIMPUL* (node) : adalah elemen dari pohon yang memungkinkan akses pada sub pohon dimana simpul tersebut berfungsi sebagai AKAR.
- *CABANG* (path): hubungan antara akar dengan sub pohon.
- *AYAH* (parent) : akar dari sebuah pohon adalah AYAH dari sub pohon.
- *ANAK* (child) : anak dari sebuah AKAR adalah sub pohon.
- *SAUDARA* (siblings) : adalah simpul-simpul yang mempunyai AYAH yang sama.
- *DAUN* : adalah simpul terminal dari pohon. Semua simpul selain daun adalah simpul BUKAN-TERMINAL.
- *JALAN* (path) : adalah suatu urutan tertentu dari CABANG
- *DERAJAT* : derajat sebuah pohon

adalah banyaknya anak dari pohon tersebut. Sebuah simpul berderajat N disebut sebagai pohon N-aire. Pada pohon biner, derajat dari sebuah simpul mungkin 0-aire (daun), 1 -aire/uner atau 2-aire/biner.

- *TINGKAT* (Level) : adalah panjangnya jalan dari AKAR sampai dengan simpul yang bersangkutan. Sebagai perjanjian, panjang dari jalan adalah banyaknya simpul yang dikandung pada jalan tersebut. Akar mempunyai tingkat sama dengan
- Dua buah simpul disebut sebagai SEPUPU jika mempunyai tingkat yang sama dalam suatu pohon.
- *KEDALAMAN* atau Tinggi sebuah pohon adalah nilai maksimum dari tingkat simpul yang ada pada pohon tersebut. Kedalaman adalah panjang maksimum jalan dari akar menuju ke sebuah daun.
- *LEBAR* sebuah pohon adalah maksimum banyaknya simpul yang ada pada suatu tingkat.

Contoh representasi pada gambar pohon sebagai berikut



Gambar 4 Contoh Pohon

Maka A dihubungkan dengan B dan C, untuk menunjukkan bahwa AKAR A dan kedua himpunan {B,D,E,F} dan {C,G,H,I} masing-masing adalah pohon dengan akar B dan C.

Di dalam pohon biner (pohon yang tiap node-nya maksimum hanya mempunyai dua anak) berlaku persamaan sebagai berikut. Jika diberikan sebuah pohon biner dengan N elemen. Dan misalkan :

- b adalah banyaknya simpul biner
- u adalah banyaknya simpul uner
- d adalah banyaknya daun

Maka akan selalu berlaku hubungan :

$$N = b + u + d$$

$$n-1 = 2b + u$$

$$b = d - 1$$

Meskipun cukup sederhana representasi pohon ini bisa menyederhanakan dan merumuskan berbagai masalah yang cukup rumit. Salah satunya adalah implementasi *Artificial Intelligence* pada suatu game.

4. Game tic tac toe

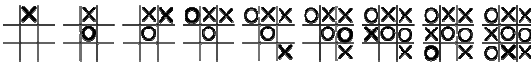
Tic tac toe adalah salah satu game klasik yang hanya bisa dimainkan oleh dua orang pemain. Kedua orang pemain itu bergiliran mengisi tanda yang berbeda (biasanya silang dan lingkaran) di dalam kotak sebesar 3 x 3. Pemain yang berhasil memposisikan tandanya secara horisontal, vertikal, atau diagonal sebagai baris yang penuh akan memenangkan pertandingan.

Contoh ilustrasinya sebagai berikut :



Gambar 5 Contoh pertandingan tic tac toe

Ilustrasi game diatas dimenangkan oleh pemain yang menggunakan tanda X.



Gambar 6 Contoh pertandingan tic tac toe

Permainan di atas berakhir seri. Jika seorang pemain sadar bahwa dirinya tidak bisa menang maka hasil seri lah yang paling baik baginya. Karena itu strategi salah satu pemain di atas adalah berusaha bertahan (defense) dengan cara menghalangi pemain lainnya untuk membentuk sebuah garis lurus.

4.1 Mengapa tic tac toe

Kesederhanaan game ini membuatnya menjadi contoh yang ideal dan mudah dipahami untuk pembelajaran konsep Combinatorial game dan *Artificial Intelligence* (kecerdasan buatan) dengan permodelan pohon. Dengan bantuan kalkulasi program komputer secara langsung

yang terimplementasi dalam program untuk permainan tic tac toe di dapat statistik sebagai berikut :

- ❖ Analisa terhadap 765 bentuk yang essensial
- ❖ 362.880 posisi akhir jika belum diperhitungkan menang atau kalahnya.
- ❖ 26.380 posisi akhir jika diperhitungkan menang atau kalahnya (memperhitungkan bentuk simetri).
- ❖ 255,168 posisi jika tidak memperhitungkan bentuk simetri, dengan rincian (jika X jalan terlebih dahulu) :
 - 131,184 game dimenangkan oleh X
 - 77,904 game dimenangkan oleh O
 - 46,080 game berakhir seri
- ❖ 31.896 kemungkinan jalannya suatu game.

Komputer *Artificial Intelligence* pertama untuk game ini adalah OXO atau Noughts and Crosses (dibuat tahun 1952) yang dibuat pada platform EDSAC dan berhasil menciptakan salah satu *Artificial Intelligence* yang mampu melawan manusia.

Salah satu contoh Komputer permainan Tic Tac Toe adalah TinkerToy computer, yang dikembangkan oleh salah satu mahasiswa MIT. Komputer ini hanya memainkan Tic Tac Toe dan belum pernah kalah sekalipun. Saat ini mesin ini dipajang di Museum of Science, Boston.

4.2 Klasifikasi game tic tac toe berdasarkan *Artificial Intelligence*

Setelah kita tahu bagaiman game ini kita bisa tinjau game tersebut dari sudut pandang *Artificial Intelligence*. Berdasarkan pembagian sifat dari sudut pandang dunia *Artificial Intelligence*, game tic tac toe mempunyai sifat seperti di bawah ini:

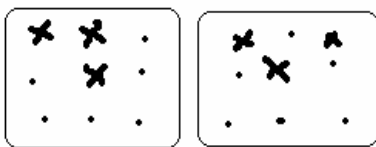
- ❖ *State-of-the-art* : algoritma berjalan di suatu komputer tertentu dan melakukan langkah yang dikalkulasi terlebih dahulu. (contoh lain : catur)

- ❖ *One-player games* (pemain melawan environment yang telah dibuat oleh computer.(contoh lain : Rubik's cube, jig-saw puzzle)
- ❖ *two person*, satu lawan satu, ada kemungkinan lawan akan menghalangi setiap strategi yang akan kita wujudkan
- ❖ *Perfect information*, pada saat tertentu kedua pemain tau seluruh kondisi dan posisi lawan.
- ❖ *Zero-sum*, jika salah satu pemain menang maka pemain yang lain kalah.

5. Strategi yang harus dirancang oleh algoritma game

Untuk menang atau mencegah kekalahan dalam game ini. Computer harus secara konsisten melakukan langkah-langkah sesuai prioritas di bawah ini dengan mendahulukan langkah dengan prioritas tertinggi.

1. menempatkan 3 buah baris diagonal, vertikal, atau horizontal.
2. menahan lawan agar tidak membentuk tiga baris yang sempurna (horizontal, vertikal maupun diagonal)
3. menciptakan strategi dengan melakukan langkah yang membuat kita mempunyai dua kemungkinan penyempurnaan baris. Beberapa pola tersebut antara lain:



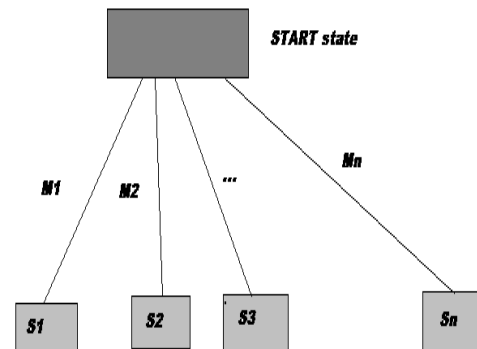
Gambar 7 pola pertandingan tic tac toe

4. Mencegah posisi lawan mempunyai pola yang bisa membuatnya menang (contoh: pola sebagaimana no.3)
5. memperbesar kemungkinan kemenangan dengan membuat dua tanda yang berdampingan.
6. mencegah lawan membuat dua tanda yang berdampingan.

Agar berhasil computer harus melengkapinya tanpa mengorbankan prioritas yang lebih tinggi secara konsisten.

5.1. Representasi pohon bagi Artificial Intelligence game ini

Kita dapat merepresentasikan seluruh kemungkinan permainan dengan graf berarah yang biasa disebut game tree (berbentuk pohon n-ary). Node dari pohon (tree) tersebut merepresentasikan keadaan game. Sisi berarah pada pohon tersebut merepresentasikan keadaan atau posisi tanda pada game.



Gambar 8 Ilustrasi Game Tree

Sebagai contoh misalkan, game dimulai pada node akar. Jika ada kondisi start tersebut terdapat N kemungkinan langkah yaitu : m_1, m_2, \dots, m_N , maka akan terbentuk sisi-sisi terhadap node state ke dua sebanyak N juga : S_1, S_2, \dots, S_N .

Dengan memperhitungkan kemungkinan langkah pada tiap tahap, kita sudah membangun suatu game tree. Daun pada pohon ini merepresentasikan kondisi akhir pada permainan. Dapat diselipkan pula beberapa nilai efisiensi pada tiap node. Sedangkan pada tiap daun diselipkan kondisi akhir pertandingan, yaitu menang kalah atau draw.

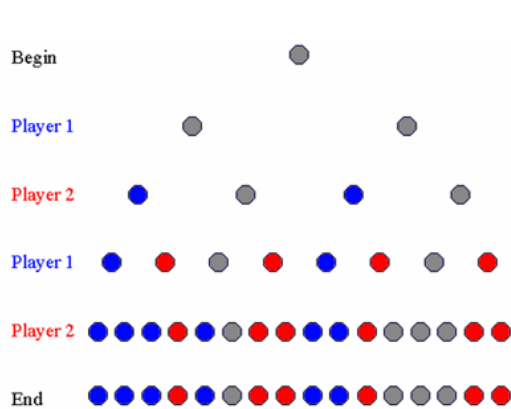
Salah satu cara untuk menciptakan *Artificial Intelligence* yang sesuai adalah dengan menganalisa seluruh game tree. Namun, sebuah game tree tic tac toe yang lengkap mempunyai 1040 nodes. Misal, kita mempunyai asumsi optimis jika suatu komputer mampu menganalisa 3 node/ nanosekon, maka akan menghabiskan waktu 1021 abad untuk menganalisa seluruh

game tree. Sehingga diciptakan berbagai macam teknik search, dan teknik representasi data.

Sebagaimana disebutkan di atas dalam teori game, sebuah game tree terdiri dari graf berarah, dimana simpulnya adalah kondisi (state) game dan sisi berarahnya adalah kemungkinan langkah.

Jumlah daun di dalam sebuah game tree yang komplit disebut kompleksitas game tree. Sedangkan untuk game tic tac toe mempunyai jumlah penyelesaian sebanyak 26.380 kemungkinan.

Meskipun terkadang tidak di kode secara langsung, pembuatan game tree sangatlah penting dalam proses pembentukan *Artificial Intelligence* dari sebuah game. Karena salah satu cara yang umum digunakan dalam pembentukan *Artificial Intelligence* sebuah game adalah menganalisis secara langsung terlebih dahulu game tree menggunakan metode algoritma *minimax* atau variasinya. Game tree tic tac toe dapat dicari dan dianalisis dengan mudah dengan menghilangkan point-point yang tidak diperlukan, namun permainan lain yang lebih besar seperti catur sangat susah untuk dianalisis secara langsung. Sehingga *Artificial Intelligence* untuk permainan seperti itu lebih cenderung ke analisis parsial dengan membagi game tree menjadi sejumlah game tree yang lebih kecil. Dengan sebuah game tree lengkap, hal itu memungkinkan menemukan langkah yang jika diikuti secara tepat terus menerus akan diketahui apakah seorang pemain menang atau tidak. Analisis untuk game tree besar seperti catur, bisa menggunakan algoritma rekursif di bawah ini:



Gambar 9 Ilustrasi Pewarnaan Game Tree

Warnai kondisi terakhir dari game tree sehingga semua kondisi yang mempunyai kecenderungan hasil akhir tertentu akan mempunyai warna berbeda. Misalkan, kondisi dimana jika pemain pertama mempunyai kemungkinan besar untuk menang maka kondisi itu ditandai dengan node berwarna biru. Sedangkan kondisi dimana pemain kedua yang mempunyai kemungkinan besar menang ditandai dengan node berwarna merah. Dan kondisi lain ditandai dengan warna abu-abu.

Lihat pada game tree di atas. Jika terdapat node yang berwarna berkebalikan (node yang menguntungkan lawan) maka langkah itu tidak disarankan untuk diambil. Sedangkan jika terdapat node di bawahnya yang berwarna sama maka langkah tersebut bisa diperhitungkan dalam pengambilan keputusan.

Diagram di atas adalah salah satu contoh penerapan game tree dengan algoritma yang tidak biasa.

6. Representasi Minimax tree

Minimax Game Tree biasanya digunakan dalam pemrograman komputer dimana permainan dilakukan dengan cara pengambilan langkah yang bergantian (bergiliran). Pada dasarnya, sebagaimana game tree biasanya merupakan representasi pohon dari semua kemungkinan langkah/gerakan. Lihat pada minimax tree di bawah. Minimax tree tersebut merepresentasikan semua kemungkinan langkah yang sudah disederhanakan. Dengan menghilangkan semua pencerminan dan rotasi dari posisi yang simetri.



Gambar 5 Minimax Tree

Perhatikan bahwa pohon di atas tidak seperti pohon n-ary biasa yang tepat memiliki nilai cabang maksimal. Node di dalam minimax tree nantinya akan mempunyai jumlah cabang (anak) sesuai kondisi dan jenis game.

Pohon di atas memiliki kedalaman tiga. Namun, di dalam pemrogramannya kedalaman dari

minimax tree hanya dianggap sebagai suatu lapisan yang berbeda. Pada setiap lapisan, terjadi pergantian pemain. Kedua pemain biasanya direpresentasikan sebagai *Max* dan *Min* (metode penamaan akan dijelaskan lebih lanjut)

Dengan sebuah minimax tree lengkap, sebuah komputer dapat memeriksa setiap langkah untuk menentukan langkah terbaik. Tentu saja seperti yang kita lihat di diagram pohon di atas mungkin mempunyai ukuran yang sangat besar dengan hanya beberapa kondisi langkah saja, dan bisa memberatkan sebuah mesin/komputer sederhana. Sehingga, untuk game berskala agak besar seperti Catur dan Go, program komputer dipaksa untuk menentukan siapa yang kalah dan menang dengan cara mempertimbangkan pembagian dari seluruh pohon. Di dalam implementasinya programmer mungkin untuk menggunakan berbagai algoritma dan trik seperti penyederhanaan Alpha-Beta, Negascout, dan MTD (f) untuk meminimalisasi jumlah node yang harus diperiksa oleh mesin atau komputer.

Minimax game tree, tentu saja, tidak dapat digunakan dengan baik pada game yang tidak bisa dilihat kemungkinan langkahnya oleh komputer. Seperti poker misalnya, komputer akan membutuhkan waktu yang lama untuk menentukan apakah langkah yang akan dilakukan oleh musuh. Karena komputer tidak mungkin melihat kartu di tangan musuh. Sehingga, minimax game tree dapat diimplementasikan dengan baik pada permainan sejenis itu. Permainan seperti ini, seperti Catur, Checkers, Othello, Igo, disebut *perfect information game*.

Alasan mengapa struktur data ini disebut minimax game tree karena kesederhanaan algoritma yang bisa langsung diterapkan pada struktur ini. Mari kita coba untuk merepresentasikan hasil dari permainan game tic tac toe dengan suatu nilai (point). Misal jika X menang, maka nilai dari permainan akan ditambah satu, sebaliknya jika O yang menang maka nilai permainan akan berkurang satu. Sehingga dari sudut pandang seperti ini X akan berusaha meninggikan nilai permainan. Sedangkan O berusaha meminimalisasi nilai point permainan. Itulah sebabnya, pada sudut pandang struktur data minimax kedua pemain diberi nama Max dan Min. Hal ini pula yang menyebabkan terlahirnya nama Minimax game tree yang diberikan pada struktur data ini.

Logika minimax ini dapat diperluas untuk game lain seperti catur misalnya. Pada game yang lebih rumit ini, program hanya dapat menganalisis sebagian dari minimax tree saja. Bahkan biasanya program tidak mengetahui akhir dari minimax tree ini. Sehingga pada kondisi ini, algoritma analisis hanya dapat dilakukan pada beberapa node saja. Kemudian komputer mencoba untuk memperkirakan siapa yang unggul pada tiap node dan perkiraan ini digunakan untuk menebak keadaan permainan. Jika komputer bermain sebagai Max, maka komputer akan berusaha menaikkan nilai point dari suatu posisi permainan. Dan skak mat (check mate) direpresentasikan sebagai nilai point maksimal dari suatu node (misalkan +1 Milyar). Jika komputer bermain sebagai Min, tentu saja akan berusaha meminimalisasi nilai point value. Nilai point value ini digunakan untuk menentukan atau menebak sisi mana yang lebih unggul.

Konsep yang sederhana dan ringkas yang diimplementasikan pada minimax ini adalah salah satu keunggulan dari minimax tree. Misalkan sebuah komputer dikondisikan untuk bermain sebagai Max. langkah terbaik adalah langkah yang mempunyai hasil point value tertinggi, dan pemain lain berusaha menurunkannya.

Pada akhirnya, kekuatan komputer bergantung pada kemampuannya untuk mengevaluasi posisi dan seberapa dalam komputer dapat membaca kemampuan manusia dalam permainan tersebut. Seorang Grand Master catur sering kali mampu untuk mengevaluasi perbedaan kecil dari permainan seorang amatir dan game master catur juga bisa melihat dan memperkirakan lebih dalam. Sehingga permodelan komputer terlihat seperti grand master tersebut yang tidak boleh bermain secara ceroboh karena hal itu akan mendapat respon yang buruk dari lawannya.

Terdapat banyak sekali algoritma yang mampu membantu merubah minimax tree menjadi lebih efisien. Salah satu diantaranya adalah penyederhanaan Alpha Beta. Pada metode ini, komputer hanya menganalisis kira-kira hanya sejumlah akar kuadrat dari node yang harus dianalisis. Misalkan, jika biasanya kita harus menganalisis 400 node, maka dengan metode ini komputer hanya perlu memeriksa 20 node saja.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. (2004). *Matematika Diskrit*. Departemen Teknik Informatika, Institut Teknologi Bandung.
- [2] Allis, L.V.1994. *Searching for Solutions in Games and Artificial Intelligence*. PhD Thesis, University of Limburg, Maastricht
- [3] Allis, L.V., M. van der Meulen, and H.J. van den Herik.1994. *Proof-Number Search. Artificial Intelligence*.
- [4]Kierulf, A.1995. *Smart Go Board: Algorithms for the Tactical Calculator*. Diploma thesis , ETH Zürich
- [5] Müller, M.1996.*Computer Go as a Sum of Local Games: An Application of Combinatorial Game Theory*. ETH Zürich
- [6] Müller, M.1996. *Playing it safe: Recognizing secure territories in computer Go by using static rules and search*.
- [7] H. Matsubara (ed.)1997. *Game Programming Workshop in Japan '97*.Computer Shogi Association. Tokyo, Japan