Penerapan Pohon Keputusan dalam Pemilihan Langkah Catur pada Chess Bot

Giant Andreas Tambunan 13519127
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519127@std.stei.itb.ac.id

***Abstract*—Permainan catur merupakan sebuah permainan papan strategi yang dimainkan oleh dua orang. Permainan catur dimainkan dalam sebuah papan catur yang terdiri dari 64 kotak (8x8) dan terbagi sama rata menjadi 32 untuk masing masing kelompok hitam dan putih. Dalam bermain permainan catur tentu ada banyak pilihan langkah yang dapat kita ambil dalam menjalankan strategi kita. Jika dikalkulasikan secara matematis, ada puluhan bahkan ribuan kemungkinan kejadian berbeda yang dapat terjadi hanya dalam satu langkah. *Chess bot* atau sering juga disebut Computer chess adalah sebuah komputer (hardware maupun software) yang berkapasitas untuk bermain catur. Di dalam makalah ini, akan dibahas bagaimana cara sebuah *chess bot* melakukan pemilihan langkah yang akan diambil dengan menggunakan decision tree (pohon keputusan) dan *minimax algorithm*.**

***Keywords*—Catur, Pohon, Pohon Keputusan, *Chess bot*, *minimax algorithm*.**

I. PENDAHULUAN

Permainan catur merupakan salah satu permainan papan paling popular di dunia. Permainan catur merupakan sebuah permainan berbasis strategi abstrak yang telah ada sejak ada ke-7. Kejuaraan atau perlombaan catur sendiri sudah ada sejak berabad abad lalu dan sudah diakui dunia pada sejak tahun 1886. Seiring berjalannya waktu, permainan catur mulai diakui dunia dan dijadikan salah satu cabang olahraga yang dilombakan dalam perlombaan olahraga.

Dengan popularitasnya dan banyaknya kejuaraan catur, munculah sebuah ide untuk membuat suatu mesin (komputer) yang dapat bermain catur yang dapat digunakan untuk latihan dan menguji kemampuan. Dengan adanya mesin tersebut seseorang tidak harus memiliki lawan main untuk dapat bermain catur. *Chess bot* pertama kali ditemukan pada pertengahan abad ke-20 pada saat era komputer tabung. Pada zaman itu *Chess bot* masih dijalankan pada super computer (pada zamannya) atau perangkat khusus, dan kualitasnya masih sangat buruk sehingga pemain pemula pun dapat mengalahkannya. Seiring perkembangan zaman, *Chess bot* terus dikembangkan hingga pada tahun 1996 *Chess bot* dapat mengalahkan pemain manusia terbaik di dunia.

Dalam permainan catur, setiap pemain memiliki 16 buah catur yang masing masing jenisnya memiliki pergerakan yang unik.Buah catur tersebut disusun di atas papan catur yang terdiri dari 64 kotak dengan posisi tertentu. Dengan memperhitungkan banyaknya buah catur, jenis buah catur, jenis pergerakan buah catur, dan kemungkinan posisi, tentunya untuk memainkan satu langkah saja memiliki banyak sekali kemungkinan kejadian. Jika dihitung secara matematis, untuk satu langkah saja (putih dan hitam menjalankan buah caturnya satu kali) akan mengasilkan puluhan bahkan ribuan kemungkinan kejadian yang berbeda.

Dengan banyaknya kemungkinan tersebut, bagaimanakah sebuah *Chess bot* memilih langkah yang harus diambil dari sekian banyak kemungkinan?

Gambar 1.1 : Catur
Sumber : https://sea.pcmag.com/news/18579/alphago-ai-conquers-top-ranked-chess-bot

II. LADASAN TEORI

Dalam menentukan langkah yang harus diambil, *Chess bot* memilih langkah dengan menggunakan pohon keputusan (decision tree) dan *minimax algorithm*.

2.1 Pohon

Pohon merupakan graf terhubung tidak berarah yang tidak mengandung sirkuit. Artinya pada suatu pohon, setiap simpul itu terhubung atau selalu memiliki lintasan yang menghubungkan suatu simpul (node) ke simpul lainnya baik itu secara langsung maupun tidak langsung. Tidak mengandung sirkuit artinya tidak ada satupun lintasan yang ada pada pohon tersebut memiliki

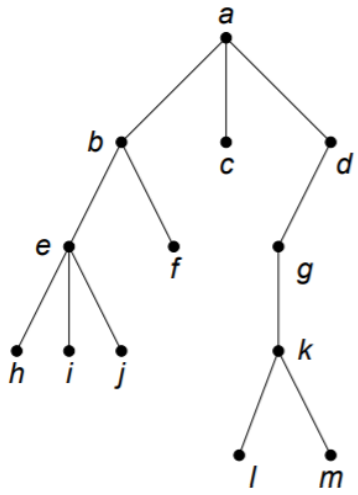
simpul awal dan simpul akhir yang sama.

Misalkan G = (V,E) dimana G adalah graf tak-berarah sederhana , V adalah himpunan simpul simpul (node) dan E adalah himpunan sisi (edge), maka untuk setiap pernyataan ini dibawah ini bernilai ekivalen :

1. G adalah pohon
2. Setiap pasang simpul (node) pada pohon G memiliki lintasan tunggal
3. G terhubung untuk semua simpulnya dan memiliki n-1 sisi (edge)
4. G tidak memiliki/mengandung sirkuit. Penambahan satu sisi pada G akan membentuk 1 sirkuit pada G.
5. G terhubung dan semua sisinya adalah jembatan

2.2 Pohon Berakar

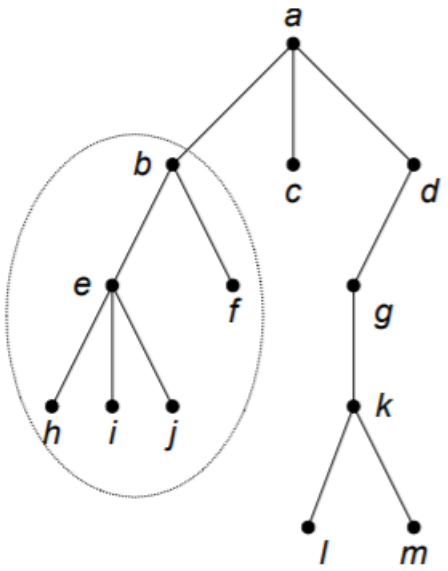
Pohon berakar atau *rooted tree* adalah pohon yang salah satu simpulnya di jadikan akar dan sisi sisinya diberikan arah.



Gambar 2.1 Pohon Berakar
Sumber :
http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf

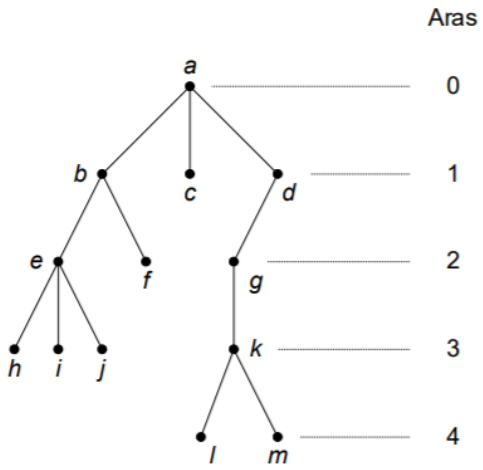
Pohon berakar memiliki terminology sebagai berikut :

1. Anak dan Orangtua
   Simpul b, c, dan d adalah anak dari simpul a dan simpul a adalah orangtua dari simpul b, c, dan d.
2. Lintasan
   Lintasan dari a ke i adala a, b, e, id an panjang lintasan dara a ke i adalah 3.
3. Saudara Kandung
   Simul yang memiliki orangtua yang sama adala simpul yang bersaudara. Misalnya simpul e adalah saudara kandung ari simpul f namun bukan saudara kandung dari simpul g.
4. Upapohon (Subtree)
   Upapohon atau *subtree* adalah sebuah pohon yang merupakan bagian dari pohon lain yang lebih besar.



Gambar 2.2 Upagraf
Sumber :
http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf

5. Derajat (*degree*)
   Derajat sebuah simpul adalah jumlah upapohon (atau jumlah anak) pohon yang terdapat pada simpul tersebut. Contohnya derajat a adalah 3, derajat b adalah 2.
6. Daun (*leaf*)
   Daun adalah simpul yang berderajat nol (atau simpul yang tidak mempunyai anak). Contohnya simpul h, i, j, f, l dan m adalah daun.
7. Simpul Dalam (*internal nodes*)
   Simpul dalam adalah simpul yang mempunyai anak dan bukan simpul akar. Contohnya b, e, g, dan k adalah simpul dalam.
8. Aras (level) atau Tingkat
   Aras suatu simpul merepresentasikan banyaknya node yang dilewati terhitung dari akar sampai ke simpul tersebut.

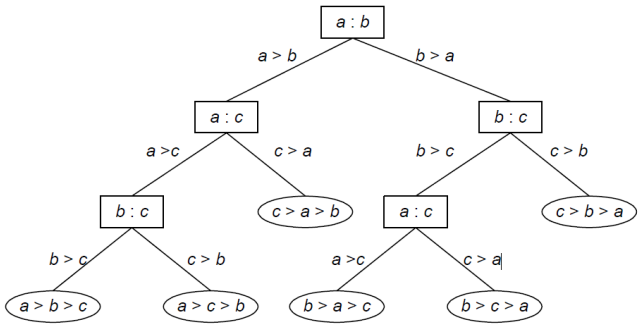


Gambar 2.3 Aras Pohon
Sumber :
http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf

9. Tinggi
   Tinggi merupakan aras maksimum dari suatu pohon. Misalnya tinggi dari pohon di atas adalah 4.

2.3 Pohon Keputusan

Pohon keputusan merupakan salah satu bentuk aplikasi dari beberapa aplikasi pohon. Pohon keputusan dapat digunakan untuk memilih keputusan yang akan diambil dengan mempertimbangkan kondisi yang ada. Berkut adalah contoh dari pohon keputusan.



Gambar 2.4 Pohon Keputusan
Sumber :
http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf

2.4 Permainan Catur

Catur adalah permainan papan strategi yang dimainkan oleh dua orang pada sebuah papan kotak-kotak yang terdiri dari 64 (8x8) kotak yang dibagi sama rata (masing masnig 32) menjadi kelopok yang berwarna hitam dan putih. Permainan ini sangatlah popular dan dimainkan oleh jutaan orang di seluruh dunia. Catur sudah ada sejak abad ke-7, bernama chaturanga dan diyakini berasal dari India. Chaturanga juga diperkirakan merupakan asal muasal permainan papan strategi yang ada di Asia Timur seperti shogi (catur Jepang), xiangqi (catur Cina) dan janggi (catur Korea).

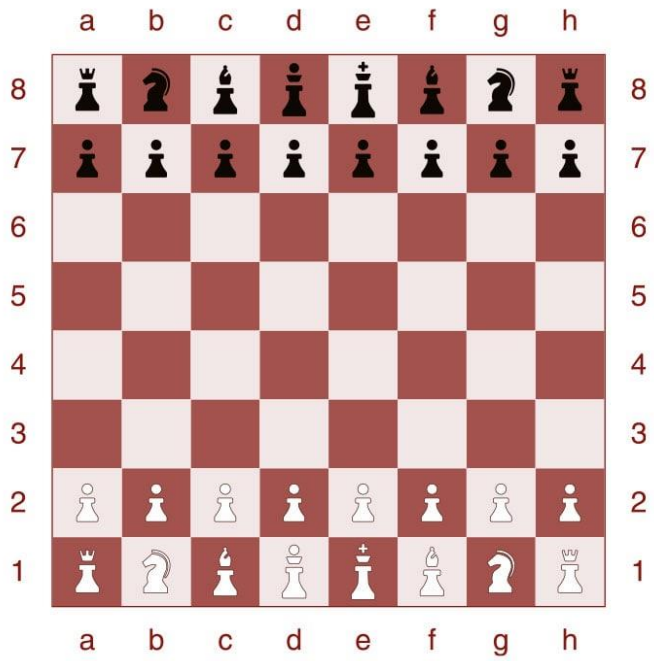
Dalam bermain permainan catur, strategi sangatlah penting. Pada mulanya setiap pemain memiliki 16 buah buah catur catur yang terdiri dari beberapa jenis yaitu: 1 raja (king), 1 menteri (queen), 2 gajah (bishop), 2 kuda (knight), 2 benteng (rook) dan 8 bidak/pion (pawn). Masing-masing dari jenis tersebut memiliki pola gerakan masing masing dan kekuatannya masing-masing. Permainan ini dilakukan secara bergantian (turn based) dan tujuan dari permainan ini adalah melakukan sekakmat.

Ada beberapa terminology pada permainan catur yang cukup penting untuk diketahui, yaitu:

1. Sekakmat (Checkmate)
   Sekakmat merupakan tujuan dari permainan ini. Sekakmat terjadi ketika raja berada pada posisi terancam dan tidak memiliki cara apapun untuk meloloskan diri dari keadaan tersebut. Pemain yang melakukan sekakmat terlebih dahulu menang.
2. Remis (Draw)
   Remis dapat diartikan sebagai seri. Permainan dapat berakhir seri dengan beberapa cara seperti: materi tidak cukup (tidak dapat melakukan sekakmat dengan sisa buah catur yang ada), pengulangan 3 langkah, kesepakatan, dsb.
3. Promosi
   Promosi terjadi ketika sebuah bidak/pion berhasil maju hingga mencapai baris akhir dan pion tersebut dapat ditukar dengan buah catur lain (ratu, gajah, kuda, benteng).
4. Raja (King)
   Merupakan buah catur yang menjadi pusat dari permainan catur. Raja harus dilindungi agar tidak disekakmat oleh lawan. Posisi awal raja berada di e1.
5. Ratu (Queen)
   Ratu merupakan buah catur yang paling kuat. Posisi awal terletak di d1.
6. Gajah (Bishop)
   Gajah merupakan buah catur yang dapat bergerak secara diagonal. Posisi awal terletak di c1 dan f1.
7. Kuda (Knight)
   Kuda adalah buah catur yang dapat bergerak dengan pola unik berbentuk L.

Posisi awal terletak di b1 dan g1.

8. Benteng (Rook)
   Benteng adalah buah catur yang dapat bergerak lurus. Posisi awal terletak di a1, dan h1.
9. Bidak/Pion (Pawn)
   Bidak adalah buah catur yang hanya bisa bergerak maju 1 atau 2 langkah kedepan. Bidak dapat memakan buah catur lawan yang berada di satu petak di samping petak yang berada tepat di depannya. Posisi awal berada di a2 sampai h2.



Gambar 2.5 Posisi Awal Catur
Sumber :
https://www.regencychess.co.uk/how_to_set_up_a_chessboard.html

2.5 *Minimax algorithm* dan Fungsi evaluasi

Minimax algorithm atau sering juga di sebut MinMax algorithm adalah algoritma aturan pengambilan keputusan yang sering digunakan dalam kecerdasan buatan, teori keputusan, teori permainan statistic dan filosofi untuk meminimalisasi kemungkinan rugi atau memaksimalkan keuntungan dalam skenario tertentu.

Secara sederhanana, *minimax algorithm* dapat diartikan sebagai suatu algoritma pencarian lintasan menuju nilai terbaik yang ada pada suatu decision tree (pohon keputusan) sampai dengan kedalaman (tinggi) tertentu. Pencarian tersebut dilakukan dengan pola tertentu dan nilai-nilai yang di maksud adalah state game (posisi dan kedudukan/dominasi) yang sudah dikalkulasi menjadi angkadengan cara tertentu (tergantung permainan yang sedang ditinjau).

Dalam teori permainan, *minimax algorithm* digunakan untuk memutuskan pengambilan keputusan untuk mendapat keputusan yang paling menguntungkan dengan banyak pertimnbangan. Dalam permainan catur sendiri, pertimbangan-pertimbangan tersebut dapat berupa posisi, nilai/banyak buah catur tersisa, state game dsb. Pertimbangan-pertimbangan tersebut kemudian akan diubah menjadi suatu nilai numerik yang dapat dibandingkan dengan suatu fungsi evaluasi. Fungsi evaluasi dalam catur sendiri terdiri dari keseimbangan materi yang mendominasi evaluasi, ditambah satu set istilah posisi yang biasanya berjumlah tidak lebih dari nilai bidak, meskipun di beberapa posisi istilah posisi bisa menjadi jauh lebih besar, seperti saat skakmat sudah dekat.

Berikut adalah *minimax algorithm* sederhana yang dituliskan dalam bentuk pseudocode

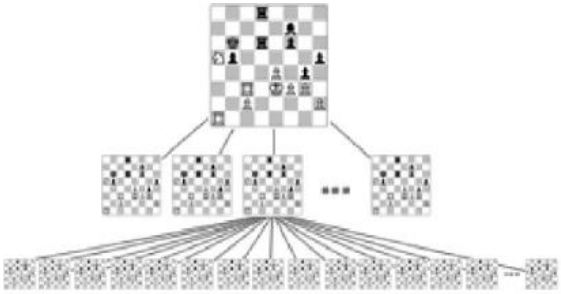
```
function minimax(node, depth, maximizingPlayer)
    if (depth = 0 or node is a terminal node) then
        return the heuristic value of node
    if (maximizingPlayer) then
        value := −∞
        for each child of node do
            value := max(value, minimax(child, depth − 1, FALSE))
        return value
    else (* minimizing player = FALSE*)
        value := +∞
        for each child of node do
            value := min(value, minimax(child, depth − 1, TRUE))
        return value
```

Gambar 2.6 : Pseudocode *Minimax algorithm*
Sumber : https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/

Untuk pembahasan lebih dalam mengenai *minimax algorithm* dan fungsi evaluasi (seperti penerapan dan source code) tidak dibahas di makalah ini dikarenakan makalah ini lebih memfokuskan implementasi pohon keputusan.

III. Penerapan Pohon Keputusan dalam Pemilihan Langkah Catur pada Chess Bot

Dalam Pengambilan keputusan langkah saat bermain permainan catur, *chess bot* merepresentasikan data state game kedalam suatu pohon keputusan. State-state game dari kemungkinan banyak kejadian berbeda diubah kedalam data berbentuk numerik dan di simpan dalam daun daun tree.



Gambar 3.1 : Ilustrasi Pohon yang Berisi kemungkinan-kemungkinan dalam permainan catur
Sumber :
https://stanford.edu/~cpiech/cs221/apps/deepBlue.html

3.1 Mengubah State Game Menjadi Angka

Sebelum bisa membuat pohon keputusan untuk memutuskan langkah yang di ambil, *chess bot* harus terlebih dahulu bisa menilai state game (keadaan permainan). Penlaian ini sangat diperlukan agar *chess bot* dapat mengenali suatu kejadian apakah kejadian itu menguntungkan atau merugikan.

Untuk itu diperlukan suatu fungsi evaluasi yang dapat mengolah data data yang ada di atas papan catur tersebut menjadi suatu angka sehingga dapat dienali oleh komputer.

Fungsi evaluasi untuk catur dibuat untuk dapat menimbang keseimbangan materi (buah catur), istilah posisi, keselamatan raja, banyaknya buah catur tersisa, mobilitas, control, keselamatan raja, pusat kendali, struktur bidak, tropisme raja, dan sebagainya menjadi suatu data angka agar dapat dikenali oleh komputer.

Fungsi evaluasi untuk catur mungkin berbentuk:

c1 * materi + c2 * mobilitas + c3 * keselamatan raja + c4 * pusat kendali + c5 * struktur bidak + c6 * tropisme raja + ...

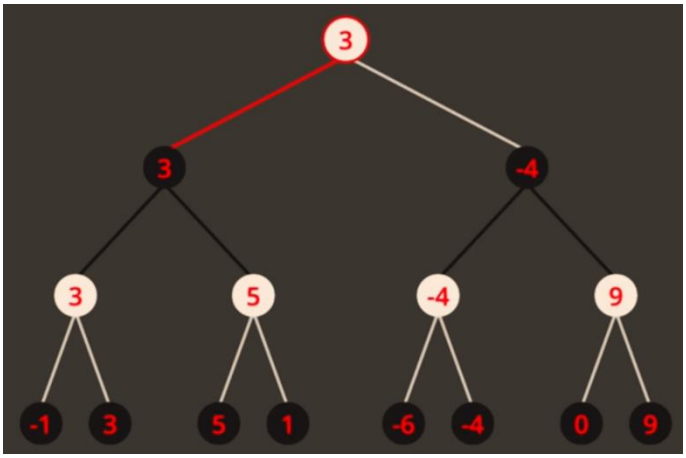
Dengan c1, c2, .. dll adalah suatu konstanta faktor yang bervariasi tergantung variable yang dibawakannya. Nilai materi bisa diperoleh dengan memberi nilai satuan kepada masing-masing jenis buah catur. Untuk yang konvensional adalah : Ratu = 9, Benteng = 5, Gajah dan Kuda = 3 dan pion = 1. Sedangkan raja diberi nilai besar yang sewenang-wenang dan biasanya lebih besar dari jumlah nilai buah catur lainnya. Selain itu rasio jumlah material yang dimiliki oleh kedua pemain juga dipertimbangkan. Skor mobilitas adalah jumlah gerakan legal yang tersedia untuk pemain, atau secara bergantian jumlah dari jumlah ruang yang diserang atau dipertahankan oleh setiap bidak, termasuk ruang yang ditempati oleh bidak lawan atau teman. Skor keselamatan raja adalah beberapa bonus penalty yang dinilai untuk lokasi raja dan konfigurasi bidak yang berada di dekat raja baik itu bidak lawan ataupun bidak milik sendiri. Skor pusat kendali diperhitungkan melalui konfigurasi dari beberapa bidak yang menempati ruang tengah papan catur. Struktur bidak adalah seperangkat hukuman dan bonus untuk berbagai kekuatan dan kelemahan dalam struktur bidak, seperti penalti untuk bidak ganda dan bidak terisolasi. Tropisme raja adalah bonus untuk kedekatan (atau penalti jarak) bidak tertentu, terutama ratu dan ksatria, dengan raja lawan. Untuk perhitungan perhitungan lain dan perhitungan yang lebih menditail mengenai fungsi evaluasi tidak dibahas lebih lanjut pada makalah ini.

3.2 Membuat Pohon

Setelah *chess bot* dapat mengenali state game, *chess bot* dapapt membuat pohon yang daunnya beriskan nilai/skor state game kemungkinan-kemungkinan kejadian dengan keadalaman (tinggi pohon) tertentu. Dalam hal ini, kedalaman pohon mewakili banyaknya langkah (baik langkah sendiri maupun langkah musuh) yang sudah dilewati.

Kedalaman pohon disini sangat diperlukan untuk menjaga performa dari *chess bot*. Misalnya kedalaman pohon yang dibuat kurang, maka *chess bot* akan mudah dikalahkan. Sebaliknya jika tidak ada batasan kedalaman di buat, maka *chess bot* akan terus mengiterasi semua kemungkinan yang ada dan membuat tree nya. Hal ini tentunya tidak akan terjadi karena semua kemungkinan kejadian pada catur sangatlah banyak yaitu sekitar 10^{120} dan tidak mungkin dapat diatasi oleh komputer. Maka dari itu semua kemungkinan kemungkinan kejadian haruslah dibuat dengan kedalaman tertentu. Hal ini juga yang menjadi alasan banyaknya tingkat kesulitan dari *chess bot*. Semakin dalam tree yang dibuat, maka akan semakin sulit *chess bot* itu untuk dilawan.

Setelah pohon dengan daunnya yang berisikan nilai/ skor state, kemudian *chess bot* akan mulai menyeleksi langkah mana yang paling menguntungkan untuk diambil dengan menggunakan *minimax algorithm*.

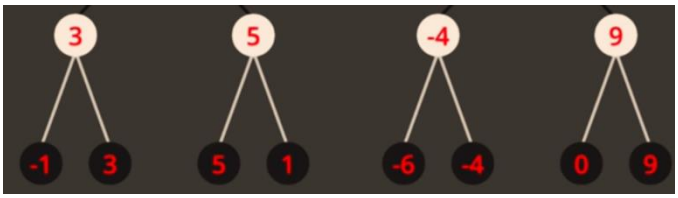


Gambar 3.2 : Contoh Pohon

Mari kita simak contoh pohon dengan kedalaman 3 yang terbentuk pada gambar 3.2 di atas. Simpul

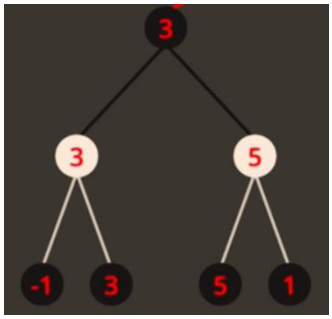
yang berwarna putih akan melambangkan nilai game state pada saat putih mengambil langkah, sebalikya simpul yang hitam melambangkan langkah hitam. Pertama tama *chess bot* akan mnecari semua kemungkinan kejadian yang dapat terjadi dalam 4 langkah dan mengkalkulasikannya ke dalam bentuk angka dan membentuk daun dari nilai tersebut. Sebut saja ada 8 kemungkinan yang dapat terbentuk seperti pada gambar.

Kemudian dengan menggunakan algoritma minimax, *chess bot* mulai menyeleksi langkah yang akan diambil agar mendapat keuntungan terbesar.



Gambar 3.3 : Perbandingan pertama

Untuk node putih (*chess bot*), bot akan mengambil nilai yang paling tinggi dari state game dibawahnya .

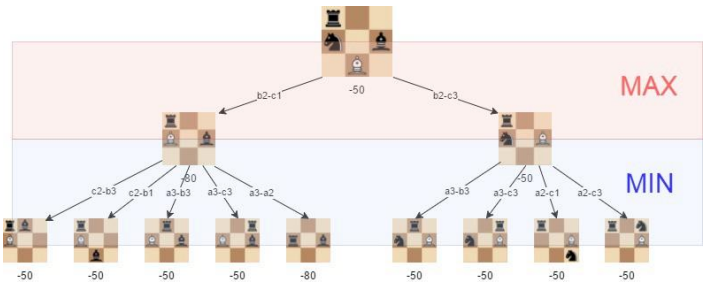


Gambar 3.4 : Perbandingan kedua

Selanjutnya, *chess bot* akan mengambil nilai terendah sebagai bentuk skor langkah unutk hitam (musuhnya). Skor terendah yang diambil merepresentasikan kondisi langkah terbaik yang akan diambil oleh hitam (musuhnya), yang mana merupakan kondisi terburuk bagi putih (*chess bot*).

Dengan menggunakan *minimax algorithm*, *chess bot* akan terus mengiterasi semua kemungkinan yang ada pada tree tersebut. Hingga akhirnya semua pohon terisi sampai ke akar. Kemudian *chess bot* akan memutuskan langkanya berdasarkan langkah yang ada.

Berikut merupakan sebuah contoh ilustrasi sederhana dari pohon keputusan yang dibentuk oleh *chess bot*.



Gambar 3.5 : Contoh ilustrasi

Sumber : https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/

Pada kasus diatas, *chess bot* (putih) akan memilih langkah kiri karena memiliki nilai yang lebih tinggi daripada yang kanan.

IV. KESIMPULAN

Aplikasi pohon seperti pohon keputusan sangatlah banyak digunakan dalam pembuatan AI (*Artificial Intelligence*) untuk game/permainan berbasis giliran dan strategi seperti catur, go, othelo, checker ataupun tic-tac-toe. Dengan mengunakan pohon, kemungkinan kemungkinan kejadian-kejadian yang dapat terjadi di dalam permainan dapat direpresentasikan sebagai data yang dapat dimengerti mesin dengan baik. Dengan menggunakan pohon juga mempermudah programmer memvisualisasikan bagaimana mekanisme yang terjadi dibalik cara kerja sebuah AI.

.

V. UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya lah penulis dapan menyelesaikan makalah yang berjudul "Penerapan Pohon Keputusan dalam Pemilihan Langkah Catur pada *Chess bot*"ini dengan baik. Penulis mengucapkan terima kasih kepada teman-teman yang selalu mensupport dalam perkuliahan ini dan semua pihak yang telah mendukung dan membantu penulis dalam menyusun makalah ini dengan baik. Penulis juga ingin mengucapkan terima kasih kepada Ibu Fariska Zakhralativa Ruskanda, S.T., M.T. selaku pembimbing dan juga dosen mata kuliah IF2120 Matematika Diskrit.

REFERENCES

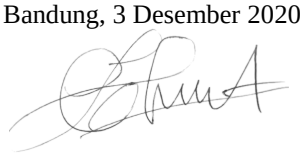
[1] https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/ , diakses pada 8 Desember 2020.
[2] Munir, Rinaldi, Slide Perkuliahan IF2120 Pohon Bagian 1, diakses pada 7 Desember 2020.
[3] Munir, Rinaldi, Slide Perkuliahan IF2120 Pohon Bagian 2, diakses pada 7 Desember 2020.
[4] http://www.cut-the-knot.org/Curriculum/Games/MixedStrategies.shtml , diakses pada 8 Desember 2020.
[5] https://stanford.edu/~cpiech/cs221/apps/deepBlue.html , diakes pada 8 Desember 2020.
[6] https://www.gamedev.net/reference/articles/article1208.asp , diakses pada 8 Desember 2020.
[7] Michael Maschler, Eilon Solan & Shmuel Zamir (2013). "Game Theory". Cambridge University Press. hal176–180.

[8] https://sea.pcmag.com/news/18579/alphago-ai-conquers-top-ranked-chess-bot , diakses pada 8 Desember 2020.
[9] https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/ , diakses pada 9 Desember 2020
[10] https://web.archive.org/web/20041103012847/http://myweb.cableone.net/christienolan/coach/evaluating.htm , diakses pada 9 Desember 2020.

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, 3 Desember 2020



Giant Andreas Tambunan
13519127