
KECERDASAN BUATAN MENGGUNAKAN ALGORITMA ALPHA –BETA PRUNING PADA PERMAINAN CATUR XIANG QI

Daniel Udjulawa
Teknik Informatika STMIK GI MDP
Jalan Rajawali, No. 14 Palembang
Pos-el : daniel@mdp.ac.id

Abstract

Artificial Intelligence is an attempt to design a machine so that it can think, take action and determine action through a computer. Like humans the decision will result in a decision taken in a game using a computer. The computer game that is applied in this study is a strategy game that is Chess Xiang qi or often called Chess Elephant or Chinese Chess. Unique features include the unique movements of the pao (cannon), the rules that prohibit the commander (such as the "king" in international chess) from meeting in the same line, and the "river" and "palace" which limit the movement of chess pieces. This research emphasizes the implementation of the Alpha Beta Pruning algorithm. Alpha Beta Pruning Algorithm is an algorithm that is used to prevent the expansion of nodes to get a better step search result than before. Alpha Beta Pruning algorithm is tested with a depth iteration process and predetermined. The methodology used is iteration. The making of this game uses the Unity game engine and the Java and C # programming languages. The results obtained in this study are to provide an overview of the application of the Alpha Beta Pruning algorithm that is used in building an artificial intelligence that exists in the chess game Xiang qi.

Keywords: artificial intelligence, Chess Xiang qi, Alpha Beta Pruning, Unity

Abstrak

Kecerdasan Buatan merupakan usaha merancang sebuah mesin agar dapat berfikir, mengambil tindakan serta menentukan tindakan melalui komputer. Layaknya manusia keputusan tersebut akan mengakibatkan sebuah keputusan yang diambil dalam sebuah permainan menggunakan komputer. Permainan komputer yang diterapkan pada penelitian ini adalah permainan strategi yakni permainan Catur *Xiang qi* atau sering disebut Catur Gajah ataupun Catur Cina. Ciri-ciri yang unik termasuklah pergerakan unik *pao* (meriam), peraturan yang melarang panglima (seperti "raja" dalam catur internasional) bertemu dalam garis yang sama, serta "sungai" dan "istana" yang membatasi pergerakan buah catur. Penelitian ini ditekankan pada implementasi algoritma *Alpha Beta Pruning*. Algoritma *Alpha Beta Pruning* adalah algoritma yang digunakan untuk mencegah perluasan node untuk mendapatkan hasil pencarian langkah yang lebih baik dari sebelumnya. Algoritma *Alpha Beta Pruning* di uji dengan proses iterasi kedalaman dan preset yang telah ditentukan. Metodologi yang digunakan adalah Iterasi. Pembuatan game ini menggunakan *game engine Unity* dan bahasa

pemograman *java* dan *C#*. Hasil yang didapat pada penelitian ini adalah memberikan gambaran tentang penerapan algoritma *Alpha Beta Pruning* yang digunakan dalam membangun sebuah kecerdasan buatan yang ada pada game catur *Xiang qi*.

Kata kunci: kecerdasan buatan, Catur *Xiang qi*, *Alpha Beta Pruning*, *Unity*

I. Pendahuluan

Perkembangan *Artificial Intelligence* banyak diterapkan pada game-game dengan menerapkan algoritma yang sesuai, sehingga akan membantu pada game tersebut dalam kegiatannya memberikan keputusan akan jalannya permainan tersebut. Sebuah permainan pada papan sederhana akan digunakan sebagai penerapan algoritma yang dituangkan dalam bentuk digital.

Salah satunya adalah catur *Xiang qi*, game ini tidaklah berbeda dengan game papan lainnya walaupun game ini lebih langka dari pada Catur Internasional. Saat ini game papan masih diminati dikalangan anak muda. Game Papan adalah game yang biasanya terdiri dari papan dan berbagai jenis potongan (dadu, bidak, counter, dll.), sistem aturan, dan dua atau lebih pemain.

Dengan adanya bantuan teknologi, game papan tersebut beralih menjadi game digital. Game Digital adalah program interaktif

untuk satu atau lebih pemain, yang merupakan adaptasi sistem game 'tradisional', dengan aturan, representasi pemain, dan lingkungan yang dikelola melalui badan elektronik, Karena keterbatasan teknologi, game digital terdahulu sangatlah jarang yang memiliki fitur multiplayer, maka diharuskan memiliki *Artificial Intelligence* atau yang disebut secara singkat dengan istilah "AI". (G.N. Yannakakis 2012)

Artificial Intelligence adalah serangkaian algoritma yang luas yang juga mencakup teknik dari teori kontrol, robotika, grafik komputer, dan ilmu komputer secara umum, AI pada game digital sering tidak merupakan "AI" yang serupa pada penelitian ilmiah karena AI pada game digital jarang memfasilitasi pembelajaran komputer atau kriteria standar lainnya, hanya merupakan perhitungan otomatis atau serangkaian respons yang telah

ditentukan dengan bantuan Algoritma-algoritma tertentu.

Algoritma adalah prosedur atau formula untuk menyelesaikan masalah berdasarkan apa yang konduktif pada urutan tindakan tertentu. Program komputer dapat dilihat sebagai gabungan berbagai algoritma yang rumit. Dalam matematika dan ilmu komputer, algoritma biasanya berupa prosedur kecil yang digunakan untuk memecahkan masalah yang berulang-ulang. Salah satu Algoritma yang sering dipakai pada *game* papan digital ialah Algoritma *Alpha-Beta Pruning*.

Algoritma *Alpha-Beta Pruning* adalah teknik optimisasi yang dapat mengurangi waktu perhitungan dengan lebih cepat. Algoritma tersebut memungkinkan sistem bekerja ke level yang lebih dalam di pohon *game* dan memotong cabang di pohon *game* yang tidak perlu dicari karena sudah ada langkah yang lebih baik. (R. Stuart, N. Peter 2018)

Algoritma *Alpha-Beta Pruning* merupakan algoritma AI yang paling sering digunakan dalam *game-game* berbentuk papan seperti catur karena algoritma menggunakan analisis

node pohon dalam memecahkan masalah.

Catur *Xiang qi* atau Catur Cina sendiri adalah *game* yang sangat populer di negara-negara Asia. Banyak yang percaya bahwa *Xiang qi* maupun Catur Barat merupakan pengembangan dari *game Chaturanga* yang merupakan catur asli India.

Bidak catur *Xiang qi* biasanya berbentuk cakram datar, memiliki nama-nama yang mirip dengan yang digunakan dalam catur barat dan tertulis di atasnya huruf Cina. Selain raja dan dua menteri, masing-masing pemain mulai dengan dua benteng, dua kuda, dua gajah, dua meriam, dan lima bidak.

Dengan pengertian tersebut tampak bahwa *Xiang qi* adalah *game* papan yang berbeda dengan catur pada umumnya, karena memiliki peraturan gerak yang berbeda, buah yang berbeda dan papan yang berbeda.

II. Landasan Teori

Catur *Xiang qi*

Catur *Xiang qi* atau Cina adalah *game* papan strategis untuk dua pemain. Ini adalah salah satu *game* papan paling populer di Cina,

dan itu milik keluarga yang sama dengan catur barat, *chaturanga*, dan *shogi*. *Game* ini adalah pertempuran dua pasukan untuk menangkap raja musuh.

Fitur khas *Xiang qi* adalah meriam, yang harus melompat untuk menangkap; aturan yang melarang raja untuk saling berhadapan secara langsung; area di papan, yang disebut sungai dan istana, yang membatasi pergerakan beberapa tokoh dan memperkuat pergerakan yang lain; dan menempatkan potongan-potongan di persimpangan garis, bukan di dalam kotak. (Yen, Shi-Jim Chen, JC Yang, TN Hsu, Shun-Chin. 2004).

Bahasa Pemrograman C#

C# (dibaca : *C sharp*) adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari kerangka *.NET Framework*. Bahasa pemrograman ini berbasis C++ dan dipengaruhi oleh fitur bahasa lain seperti *Java*, *Delphi*, *Visual Basic*, dan lain-lain.

Java

Sejarah Java, Java dalam ilmu komputer, merupakan bahasa

pemrograman berorientasi objek yang diperkenalkan pada tahun 1995 oleh *Sun-Microsystem Inc.*, yang saat Java diciptakan, dipimpin oleh *James Gosling*.

Unity

Unity Technologies dibangun di tahun 2004 oleh David Helgason, Nicholas Francis dan Joachim Ante. *Game engine* ini dibangun dengan *indie developer* sebagai prioritas utama karena banyak yang tidak bisa membeli *game engine* yang terlalu mahal. Fokus perusahaan *Unity* adalah membuat sebuah perangkat lunak yang bisa digunakan oleh semua orang, khususnya untuk membangun sebuah *game*. Di tahun 2009, *Unity* diluncurkan secara gratis dan di april 2012, *Unity* mencapai popularitas tinggi dengan lebih 1 juta *developer* terdaftar di seluruh dunia. (Rickman Roedavan, 2014, 5).

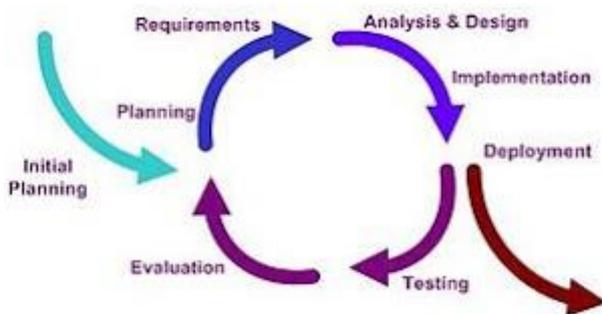
Algoritma Alpha Beta Pruning

Algoritma Alpha Beta Pruning merupakan algoritma pencarian yang digunakan untuk mengurangi jumlah node melalui proses evaluasi. Sama halnya dengan Algoritma MinMax, Algoritma *Alpha Beta Pruning* merupakan algoritma untuk mencari

solusi yang optimal dari sebuah permasalahan, namun perbedaannya ada pada proses evaluasi yang digunakan untuk mengurangi jumlah node solusi yang dibentuk pada iterasi sebelumnya.

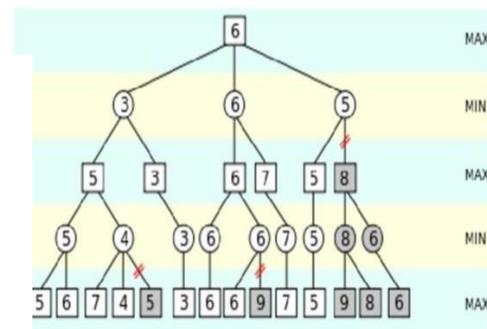
III. Metodologi Penelitian

Dalam penelitian ini penulis menggunakan metode iterasi. Iterasi adalah model pengembangan system yang bersifat dinamis dalam artian setiap tahapan proses pengembangan system dapat diulang jika terdapat kekurangan atau kesalahan. Setiap tahapan pengembangan system dapat dikerjakan berupa ringkasan dan tidak lengkap, namun pada akhir pengembangan akan didapatkan system yang lengkap pada pengembangan system. Metode Iterasi dapat dilihat pada gambar 1, dengan tahapan-tahapannya.



Gambar 1. Metode Iterasi

Algoritma *Alpha Beta Pruning* memiliki dua nilai, yaitu alpha dan beta. Alpha merupakan nilai maksimum yang dimiliki oleh pemain, sedangkan nilai beta adalah nilai minimum yang dimiliki oleh lawannya. Nilai alpha adalah negatif tak terhingga pada kondisi awal permainan dan beta adalah positif tak terhingga. Pemilihan node kemungkinan pergerakan langkah akan selalu diperbarui pada setiap kali pemain selesai memindahkan pionnya. Jika terdapat sebuah node dengan nilai beta kurang dari atau sama dengan alpha, maka node tersebut tidak akan dipilih untuk perhitungan langkah berikutnya sehingga cabang di bawah pada node itu akan dipangkas (*prune*). Adapun ilustrasi dari Algoritma *Alpha Beta Pruning* tersebut digambarkan pada Gambar 2.



Gambar 2. Algoritma *Alpha Beta Pruning*

Pada akar pohon pencarian, mula mula nilai α diset sama dengan $-\infty$ sedangkan untuk nilai beta akan diset sama dengan $+\infty$. Simpul-simpul yang melakukan proses maksimisasi akan memberikan nilai alpha dari nilai anak anaknya sedangkan simpul-simpul yang melakukan proses minimisasi akan Maksimisasi A -2 B -6 C -2 D -4 E 9 F -6 G 0 H 0 I -2 J -4 K -3 Minimisasi memberikan nilai beta dari nilai anak-anaknya.

Jika $\alpha > \beta$, maka evaluasi dihentikan. Tiap tiap simpul yang melalui nilai alpha dan beta akan dievaluasi anakanaknya. Simpul simpul anak akan memperbaiki nilai alpha dan beta. Simpul yang melakukan minimisasi akan memberikan nilai beta sesuai dengan nilai simpulnya. Sedangkan simpul yang melakukan maksimisasi akan memberikan pertama kali akan dilakukan penelusuran ke simpul B. penelusuran akan dilakukan untuk anak-anak B terlebih dahulu. Jika N bernilai tiga maka nilai tiga ini akan memperbaiki nilai beta.

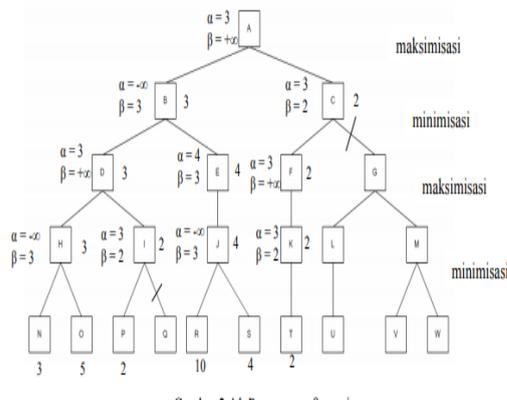
Sehingga nilai beta dari $+\infty$ menjadi tiga. Kemudian nilai ini dibandingkan dengan nilai dari O. Karena nilai O lebih besar dari N

maka nilai O tidak memperbaiki nilai beta sehingga nilai beta akan tetap menjadi tiga dan nilai H menjadi tiga. maksimisasi maksimisasi minimisasi minimisasi $\alpha = -\infty \beta = +\infty$
 $\alpha = -\infty \beta = +\infty \alpha = -\infty \beta = +\infty \alpha = -\infty \beta = 3 \ 3 \ 5 \ 3$ Dari nilai beta yang diperoleh untuk simpul N dan O akan memperbaiki nilai H sehingga H menjadi bernilai tiga dan alpha menjadi tiga. Gambar 2.4b. Penerapan $\alpha\beta$ pruning Jika P bernilai dua maka nilai ini akan dibandingkan dengan nilai beta.

Maka nilai P ini akan memperbaiki nilai beta sehingga nilai beta sama dengan 2. Karena nilai alpha lebih besar dari beta maka proses dihentikan dan cabang yang lain untuk simpul yang sama dipotong. Ini dikarenakan pada saat D melakukan proses maksimisasi maka akan dipilih simpul H karena simpul H bernilai lebih besar dibandingkan dengan simpul I. Jika dimisalkan simpul Q bernilai 4, maka nilai dari simpul I tetap dua.

Jika simpul Q bernilai satu maka nilai I malah menurun menjadi satu. Berapapun nilai dari Q tidak akan berpengaruh. Sehingga nilai D menjadi tiga. minimisasi maksimisasi maksimisasi minimisasi $\alpha = -\infty \beta =$

$+\infty \alpha = -\infty \beta = +\infty \alpha = 3 \beta = +\infty 3 5 3$
 $\alpha = 3 \beta = 2 2 2 3 \alpha = -\infty \beta = 3$.
 Penerapan $\alpha\beta$ pruning Saat simpul R bernilai 10 maka nilai ini tidak akan memperbaiki nilai beta. Begitu pula dengan nilai S yang bernilai empat tidak akan memperbaiki nilai beta sehingga beta tetap bernilai tiga. Nilai R dan S akan di bandingkan kemudian akan dicari yang terkecil yang digunakan untuk memperbaiki nilai J sehingga nilai J menjadi empat. Nilai J akan memperbaiki nilai alpha dari $-\infty$ menjadi empat. Sehingga nilai E menjadi empat dan nilai B sama dengan tiga. maksimisasi maksimisasi minimisasi minimisasi $\alpha = -\infty \beta = +\infty \alpha = -\infty \beta = 3 \alpha = 3 \beta = +\infty \alpha = -\infty \beta = 3 3 5 3 \alpha = 3 \beta = 2 2 2 10 4 \alpha = -\infty \beta = 3 \alpha = 4 \beta = 3 4 3 4 3$ Lihat Gambar 3.



Gambar 3. Penerapan Alpha Beta Pruning

Penerapan $\alpha\beta$ pruning Nilai T akan memperbaiki nilai beta yang tadinya bernilai tiga menjadi dua. Nilai ini akan memperbaiki nilai K sehingga K akan bernilai dua dan nilai F menjadi dua juga. Kemudian nilai ini diteruskan ke simpul C sehingga C bernilai dua. Simpul G akan dipotong karena nilai alpha > beta.

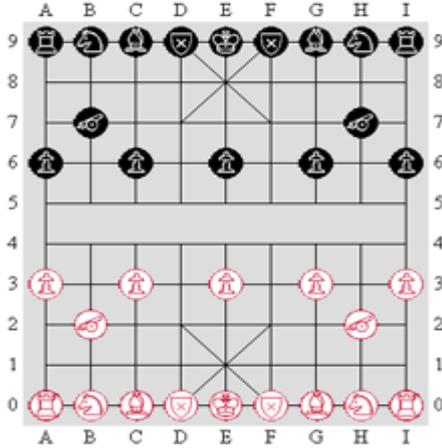
IV. Hasil Dan Pembahasan

Pada bagian ini hal yang dibahas adalah langkah permainan catur *Xiang qi* serta penerapan Algoritma *Alpha Beta Pruning*.

Permainan Catur Xiang

Permainan yang digunakan untuk penerapan algoritma ini adalah Catur *Xiang qi* atau Catur Cina. Catur ini merupakan *Game* yang menggambarkan pertempuran dua asukan untuk menangkap raja musuh. Fitur khas *Xiang qi* adalah meriam, yang harus melompat untuk menangkap, aturan yang melarang raja untuk saling berhadapan secara langsung, area di papan, yang disebut sungai dan istana, yang membatasi pergerakan beberapa tokoh dan memperkuat pergerakan yang lain dan menempatkan potongan-potongan di persimpangan garis,

bukan di dalam kotak. Lihat gambar 4..



Gambar 4. Papan Catur *Xiang qi*

Xiang qi dimainkan di papan sembilan garis lebar dan sepuluh garis panjang. Seperti dalam *game Go*, potongan ditempatkan di persimpangan, yang dikenal sebagai titik. Garis vertikal dikenal sebagai file (kolom), dan garis horizontal dikenal sebagai peringkat (baris).

Berpusat di papan peringkat pertama ke ketiga dan kedelapan hingga kesepuluh adalah dua zona, masing-masing tiga poin dengan tiga poin, dibatasi oleh dua garis diagonal yang menghubungkan sudut-sudut yang berlawanan dan berpotongan di titik tengah.

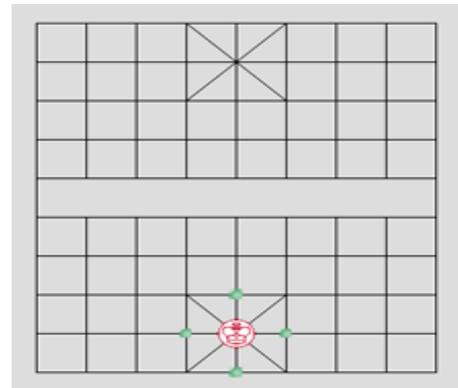
Masing-masing daerah ini dikenal sebagai kastil. Membagi dua sisi yang berlawanan, antara peringkat kelima dan keenam, adalah "sungai". Sungai ini sering ditandai

dengan frasa "Sungai *Chu*" dan "Perbatasan *Han*", merujuk pada Perang *Chu-Han*. Meskipun sungai memberikan pembagian visual antara kedua belah pihak, hanya dua potong yang terpengaruh oleh kehadirannya: tentara memiliki gerakan yang ditingkatkan setelah menyeberangi sungai, dan gajah tidak dapat menyeberanginya.

Titik awal tentara dan meriam biasanya, tetapi tidak selalu, ditandai dengan silang kecil.

Peraturan Pergerakan:

A. Raja



Gambar 5. Ilustrasi Gerakan Raja

1. Seorang raja hanya bergerak satu ruang secara vertikal atau horizontal, tidak dapat bergerak secara diagonal.
2. Tidak dapat meninggalkan istananya.
3. Seperti dalam Catur standar, seorang raja tidak diizinkan untuk

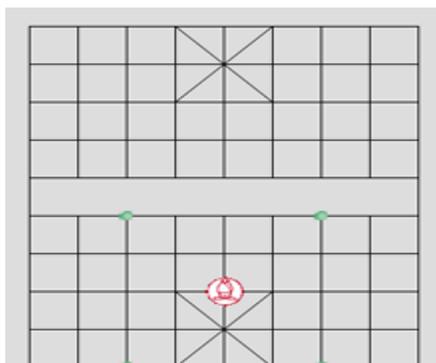
bergerak ke posisi yang sedang diserang oleh bagian musuh.

4. Jika seorang raja diserang (sedang dalam cek), pemain yang bersangkutan harus membatalkan dan cek segera, Jika tidak memungkinkan, pemain dinyatakan kalah.

B. Menteri

1. Seorang penjaga bergerak satu ruang secara diagonal dan seperti raja, ia mungkin tidak pernah meninggalkan istananya.
2. Dengan kata lain, setiap penjaga hanya memiliki 5 posisi yang memungkinkan di dalam area istana.

C. Gajah



Gambar 6. Ilustrasi Gerakan Gajah

Gajah dapat bergerak tepat dua ruang secara diagonal dan mereka tidak dapat melewati titik yang diduduki. Selain itu, gajah tidak diperbolehkan menyeberangi sungai, mereka harus tetap di atas papan. Ini

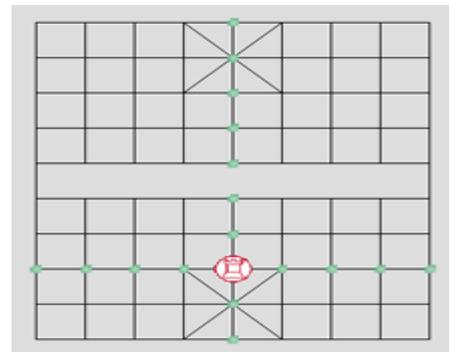
berarti setiap gajah hanya memiliki 7 posisi yang memungkinkan.

D. Kuda

Kuda bergerak satu ruang secara vertikal atau horizontal, diikuti oleh satu ruang keluar secara diagonal. Ada perbedaan yang signifikan antara Catur barat dan *Xiang qi* karena kuda tidak dapat melompati tempat-tempat yang diduduki.

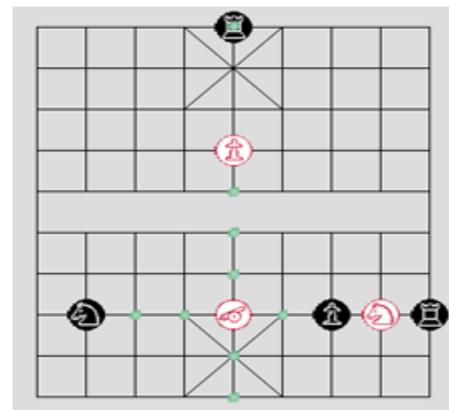
E. Benteng

Benteng bergerak sebagai benteng dalam Catur standar - sejumlah ruang secara vertikal atau horizontal



Gambar 7. Ilustrasi Gerakan Benteng

F. Meriam



Gambar 8. Ilustrasi Gerakan Meriam

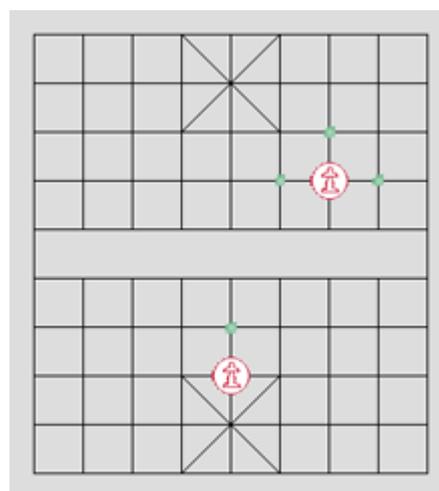
1. Meriam bergerak dengan cara yang sama seperti benteng. Namun, jika sebuah meriam ingin menangkap kepingan lawan, itu harus dilakukan dengan berharap satu kepingan lain baik milik sendiri atau milik lawan.
2. Gambar menunjukkan posisi di mana semua meriam merah yang mungkin bergerak atau menangkap ditandai dengan titik-titik hijau. Ksatria hitam di sisi kiri tidak dapat ditangkap karena tidak ada bagian ketiga berdiri di antara itu dan meriam.
3. Benteng hitam atas dapat ditangkap karena meriam melompat di atas bidak merah.
4. Di sisi lain, benteng hitam kedua di sebelah kanan tidak dapat ditangkap karena meriam akan melompati lebih dari satu bagian.

G. Bidak

1. Bidak atau Pion bergerak hanya satu ruang ke depan dan gerakan menangkap mereka hanya lurus.
2. Ketika Bidak atau pion melintasi sungai, kemampuan Bergeraknya

melebar dengan bergerak satu ruang ke depan atau horizontal.

3. Bidak atau Pion tidak berpromosi di *Xiang qi*.
4. Ketika sebuah Bidak atau pion mencapai baris terakhir, dan hanya dapat terus bergerak ke samping saja.



Gambar 9. Ilustrasi Gerakan Bidak

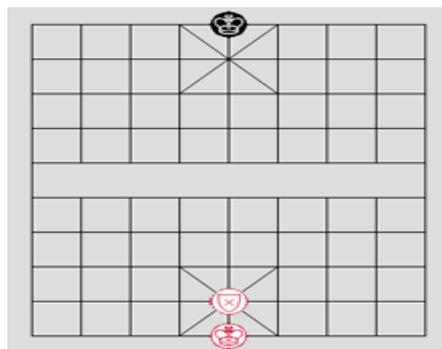
Peraturan Khusus:

1. Sebuah *game* dimenangkan dengan melakukan *checkmating* atau membuat kebuntuan raja lawan. Ini adalah perbedaan penting antara *game* ini dan Catur barat di mana jalan buntu menghasilkan hasil seri.
2. Cek abadi dilarang. Seorang pemain tidak dapat memeriksa raja lawan lebih dari 3 kali berturut-turut dengan bagian yang sama dan posisi yang sama.

3. Mengejar pion musuh yang tidak dilindungi juga dilarang. Ini berarti bahwa seorang pemain tidak dapat menyerang bagian musuh yang sama yang tidak dilindungi oleh bagian musuh lain.

Pemain yang menyebabkan situasi seperti itu berkewajiban untuk menghentikannya.

4. Raja tidak bisa saling berhadapan di kolom terbuka yang sama, harus ada setidaknya satu bagian di antara mereka di kolom yang sama. Gambar di bawah ini menampilkan posisi di mana penjaga merah tidak bisa bergerak karena raja akan terungkap satu sama lain:

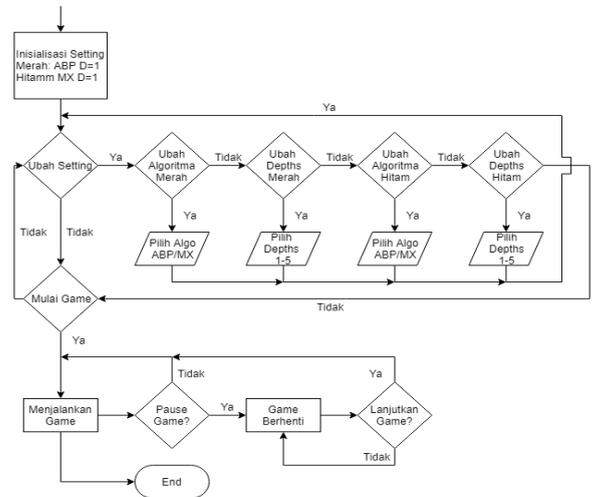


Gambar 10 . Ilustrasi Raja Tidak Boleh Saling Menatap

5. Ketika tidak ada pihak yang bisa memaksa skakmat maka *game* disebut seri.

Flowchart Game

Untuk menggambarkan urutan-urutan proses pada system ini, maka digunakan sebuah flowchart sebagai penjelas. berikut ini adalah flowchart yang menggambarkan mulai game.



Gambar 11 Flowchart Game

Dengan terkumpulnya data performa *algoritma Alphabeta* pada *Preset* papan *Xiang qi*, maka tabel dapat disusun sesuai algoritma, sehingga didapat hasil rata-rata pada kecepatan dan jumlah putaran, seperti pada tabel berikut:

Tabel 1. AlphaBeta Sisi Hitam

Black Alphabeta	1	2	3	4	5
Default	0.047	0.547	17.269	127.120	
Pawn Off The River A	0.005	0.182	2.297	46.040	
Pawn Off The River B	0.005	0.030	0.144	5.019	
Cannons Ommited	0.003	0.102	1.503	9.140	
Flanking Cannons	0.017	1.088	3.328	75.360	
Cannons and Forts	0.011	0.166	7.762	64.840	
Knights and Elephants	0.002	0.038	0.226	87.910	
Castle Siege A	0.004	0.157	1.345	26.473	
Castle Siege B	0.012	0.150	1.166	39.860	
Mirrored Advances	0.032	0.152	3.394	79.870	
Average	0.014	0.261	3.843	56.163	

Tabel 2. AlphaBeta Sisi Merah

Red Alphabeta	1	2	3	4	5
Default	0.045	0.732	17.747	190.600	
Pawn Off The River A	0.012	0.357	6.169	95.850	
Pawn Off The River B	0.011	0.069	0.756	21.427	
Cannons Ommited	0.004	0.104	1.367	8.669	
Flanking Cannons	0.015	0.404	10.699	219.190	
Cannons and Forts	0.011	0.338	16.102	116.720	
Knights and Elephants	0.004	0.050	0.327	5.695	
Castle Siege A	0.007	0.212	4.054	54.880	
Castle Siege B	0.023	0.344	3.060	55.070	
Mirrored Advances	0.056	0.377	4.660	57.140	
Average	0.019	0.299	6.494	82.524	

Tabel 3 Rata-rata Putaran AlphaBeta

Turns Alphabeta	1	2	3	4
Default	18	10	9	10
Pawn Off The River A	3	3	3	3
Pawn Off The River B	12	5	5	11
Cannons Ommited	7	7	7	7
Flanking Cannons	6	6	3	3
Cannons and Forts	4	3	5	3
Knights and Elephants	5	5	5	10
Castle Siege A	5	7	8	6
Castle Siege B	9	9	7	19
Mirrored Advances	14	3	3	4
Average	8.3	5.8	5.5	7.6

Tabel 4 Rata-rata Kecepatan Alphabeta

Average	1	2	3	4
Red Alphabeta	0.019	0.299	6.494	82.524
Black Alphabeta	0.014	0.261	3.843	56.163

Tampaklah Hasil rata-rata dengan kedalaman 4 makin membutuhkan waktu komputasi yang cukup besar.

V. Kesimpulan

Catur *Xiang qi* merupakan permainan catur yang diharapkan dapat membantu pemain dalam berlatih bermain melawan komputer. Kecerdasan buatan yang diimplementasikan pada permainan ini adalah Algoritma *Alpha Beta Pruning*.

Dari hasil pengujian yang telah dilakukan, Algoritma *Alpha Beta Pruning* dapat diimplementasikan dengan baik pada game Catur *Xiang qi*, di komputer. Kecerdasan buatan yang dibentuk akan memiliki jumlah menang lebih banyak dibandingkan dengan pemain, sehingga kecerdasan buatan yang dibangun dengan menggunakan algoritma ini dapat digunakan dalam berlatih bermain catur *Xiang qi*.

Saran

Saran dari penelitian ini ditujukan bagi para pengembang permainan catur *Xiang qi*. Ada banyak algoritma optimasi yang dapat digunakan dalam membangun kecerdasan buatan dalam permainan catur. Salah satunya adalah Algoritma Alpha Beta Pruning. Algoritma Alpha Beta Pruning memodelkan setiap kemungkinan pergerakan menjadi cabang-cabang pohon untuk mencari solusi terbaik dari masing-masing kemungkinan pergerakan yang terjadi.

Semakin banyak pergerakan langkah yang dicari kemungkinan terbaiknya, maka akan semakin besar tingkat komputasi dari komputer yang digunakan. Hal tersebut dapat ditunjukkan oleh kemampuan komputer dalam menggerakkan bidak nya. Untuk pengembangan selanjutnya mungkin Algoritma ini dapat diterapkan pada permainan lain yang mungkin lebih banyak dan besar dalam proses komputasinya.

DAFTAR RUJUKAN

G.N. Yannakakis. 2012, *Game AI Revisited*. Center for Computer

Games Research IT University of Copenhagen, Denmark.

Hidayatullah, Priyanto, Jauhari Khairul Kawistara, 2014, Pemrograman WEB. Bandung : Informatika Bandung. (jQuery)

R. Stuart, N. Peter. 2018, *Artificial Intelligence: A Modern Approach*, Pertinence Hall. New Jersey.

Yen, Shi-Jim Chen, JC Yang, TN Hsu, Shun-Chin, 2004, *Computer Chinese Chess. ICGA journal*. 27. 3-18. 10.3233/ICG-2004-27102. Taiwan.

Funk, J. B. (2002). *Electronic Games, In Strasburger and B. Wilson(Eds)*, Children, adolescents, and the media(pp. 117-144). Thousand Oaks, CA:Sage.

Desiani, A., & Arhami, M. (2006). *Konsep Kecerdasan Buatan*, Andi Offset, Yogyakarta

Camacho, MC. (2009). *Adaptive Behavior in Artificial Intelligence*, Helsinki Metropolis University of Applied Sciences.

Lestari, J., & Amalia, S. A. (2013). Implementasi Algoritma Alpha-Beta Pruning pada Permainan Bantumi dengan Berbasis Mobile Android, 10(1), 23–32. Jakarta, Universitas Budi Luhur.

Jogiyanto, H.M. (2005). *Analisis & Desain*, Ed ke-III, Andi Offset, Yogyakarta.

Claude, E. Shannon. (1950). *Programming a Computer Playing Chess Philosophical Magazine*, Ser 7, Vol 41, No. 312