Kecerdasan Buatan Berbasis Monte-Carlo Tree Search untuk Permainan Shogi pada Android

C. Pickerling[1], Hendrawan Armanto[2], Daniel[3]

Abstrak— **Permainan Shogi adalah permainan yang mulai digemari oleh masyarakat akhir-akhir ini. Untuk memainkannya, tentu diperlukan papan permainan khusus beserta bidak-bidaknya. Namun kepopuleran Shogi yang masih belum terlalu tinggi di luar negara Jepang, menyebabkan sulitnya untuk mencari papan permainan Shogi. Dikarenakan masalah tersebut, maka muncul lah permainan Shogi digital, dimana untuk bermain Shogi tidak diperlukan lagi papan permainan secara fisik, melainkan dapat dengan mudah dimainkan pada perangkat Android. Penelitian ini bertujuan untuk membuat kecerdasan buatan yang dapat memainkan permainan Shogi pada Android. Adapun kecerdasan buatan yang digunakan adalah kecerdasan buatan berbasis Monte-Carlo Tree Search. Manfaat dari aplikasi ini adalah agar seseorang dapat bermain shogi tanpa harus menggunakan papan permainan fisik dan sekaligus untuk menyediakan lawan bermain. Setelah melalui berbagai proses uji coba, dapat disimpulkan bahwa MCTS hanya mampu bersaing dengan kecerdasan buatan level easy hingga medium, namun keunggulan tersebut akan menurun drastis jika ditandingkan dengan kecerdasan buatan level hard. Akan tetapi apabila dibandingkan dengan algoritma game playing seperti Negamax AB Pruning maka MCTS dapat memenangkan rata-rata 80% pertandingan.**

Kata Kunci: **Shogi, Kecerdasan Buaran, dan Monte Carlo Tree Search.**

Abstract— **Shogi is a game that start to popular with many people these days. To play it, you need a special game board and the pieces. However, the popularity of Shogi is still not very high outside Japan, making it difficult to find a Shogi board game. Due to these problems, a digital Shogi game emerged, where to play Shogi no longer needed a physical game board, but it can be easily played on an Android device. This research aims to create artificial intelligence that can play Shogi on Android. The artificial intelligence used is based on Monte Carlo Tree Search. The benefit of this application is one can play Shogi without having to use a physical game board and at the same time provide an opponent. After various processes through trials, it can be said that MCTS is only able to compete with artificial intelligence at easy to medium level, but this advantage will decrease drastically when compared with artificial intelligence at hard level. However, when compared to game algorithms such as Negamax AB Pruning, MCTS can win an average of 80% of matches.**

Keywords: Shogi, Artificial Intelligence, and Monte Carlo Tree Search.

I. PENDAHULUAN

Shogi adalah salah satu permainan papan mirip catur yang mulai digemari oleh masyarakat umum. Seperti hal nya catur, shogi merupakan permainan dimana terdapat 2 individu yang menjalankan bidak secara bergiliran dengan tujuan untuk menangkap bidang raja lawan. Permainan ini banyak dimainkan untuk bersenang-senang akan tetapi tidak banyak juga yang memainkannya untuk mengasah orak atau bahkan bertanding. Meskipun mirip dengan permainan catur, shogi memiliki beberapa perbedaan mendasar seperti susunan dan pergerakan bidak. Hal tersebut menyebabkan kompleksitas yang berbeda sehingga membutuhkan strategi yang berbeda dalam memenangkannya, ditambah lagi dalam permainan shogi, pemain dapat meletakan kembali bidak yang ditangkap sehingga perkembangan permainan menjadi semakin sulit ditebak.

Di luar Jepang sediri, permainan shogi masih belum mencapai titik yang sangat popular akan tetapi saat ini mulai banyak orang memainkannya. Hal ini memberi dampak negatif, dikarenakan ketika seseorang hendak bermain shogi maka dia harus memesan papan permainan secara online atau membeli nya di Jepang langsung. Padahal kita ketahui sendiri, bahwa membeli dari Jepang langsung akan membutuhkan biaya yang tidak sedikit. Namun dengan berkembangnya teknologi saat ini, menyebabkan shogi dapat dimainkan dimanapun. Kita tidak perlu lagi memesan papan shogi asli melainkan dengan sebuah komputer atau smartphone, pemain dapat memainkan shogi kapanpun baik melawan pemain lain atau melawan AI.

Walaupun demikian, perkembangan AI sendiri dalam permainan shogi belum sebaik perkembangan AI pada catur. Hal ini dapat dilihat dengan banyaknya kompetisi AI catur di berbagai website catur akan tetapi hampir tidak pernah ditemukan kompetisi AI shogi. Melihat hal tersebut penelitian ini bertujuan untuk mengembangkan AI Shogi yang baik dan cepat.

II. SHOGI

Pada sub bab ini, penulis akan menjelaskan secara

[1] *Program Studi Informatika Fakultas Sains dan Teknologi Institut Sains dan Teknologi Terpadu Surabaya, Jl. Ngagel Jaya Tengah 73-77, Surabaya 60284 INDONESIA (email: pickerling@stts.edu)*

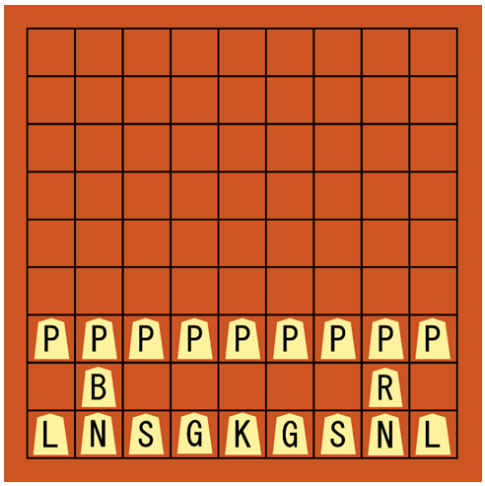
[2] *Program Studi Informatika Fakultas Sains dan Teknologi Institut Sains dan Teknologi Terpadu Surabaya, Jl. Ngagel Jaya Tengah 73-77, Surabaya 60284 INDONESIA (email: hendrawan@stts.edu)*

[3] *Program Studi Informatika Fakultas Sains dan Teknologi Institut Sains dan Teknologi Terpadu Surabaya, Jl. Ngagel Jaya Tengah 73-77, Surabaya 60284 INDONESIA (email: danielstelar77@gmail.com)*

singkat sejarah dan tata cara permainan shogi terlebih dahulu, mengingat penelitian ini bertujuan mengembangkan AI shogi yang baik maka sebaiknya pembaca memiliki konsep permainan shogi terlebih dahulu.

A. Permainan Shogi

Shogi atau yang biasa dikenal dengan nama Catur Jepang atau Permainan Jedral, adalah permainan papan untuk 2 orang dan termasuk ke dalam permainan strategi yang sekelompok dengan permainan catur dari Barat, janggi dari Korea, serta xiangqi dari Cina. Permainan ini diperkirakan berasal dari permainan India kuno yang disebut dengan caturangga. Di seluruh dunia, shogi diperkirakan menempati urutan ketiga dalam jumlah pemain setelah catur dan xiangqi. Shogi sendiri terdiri dari kata (shō 将) yang berarti Jendral, dan (gi 棋) yang berarti permainan papan. Dalam permainan shogi, kedua pemain akan bermain dengan menggunakan papan berukuran 9 baris x 9 kolom dan berwarna sama untuk setiap kotaknya. Setiap pemain akan memiliki 20 bidak, yang menghadap ke arah lawan. Di sisi kanan masing-masing pemain juga terdapat tempat bidak shogi (komadai) untuk meletakkan bidak lawan yang tertangkap. [1][2][3]



Gambar. 1. Penataan Bidak pada Papan Shogi

Terdapat 8 bidak pada shogi yang ditempatkan sesuai dengan gambar 1. Kedelapan bidak tersebut antara lain: Pawn, Rook, Bishop, Lance, Knight, Silver General, Gold General, dan King. Bidak-bidak tersebut akan bergerak menurut aturan masing-masing.

B. Aturan Khusus

Permainan shogi memiliki beberapa aturan khusus yang membedakan dengan permainan sejenis lainnya (catur atau yang lain). Aturan khusus tersebut antara lain:

1) Promotion

Di dalam catur barat terdapat aturan dimana jika sebuah pion mencapai ujung dari papan permainan, maka pion tersebut dapat ditukarkan dengan bidak lainnya (promosi). Di dalam shogi pun terdapat aturan yang mirip, dimana apabila sebuah bidak mencapai posisi tertentu di papan permainan maka bidak dapat dipromosikan. Akan tetapi berbeda dengan catur dimana hanya pion saja yang berhak memperoleh promosi, pada shogi hampir semua bidak (kecuali raja dan gold general) dapat dipromosikan. Kelebihan dari sebuah promosi pada shogi adalah cara gerak bidak akan berubah dan tidak mengikuti aturan dasar bidak.

2) Dropping Piece

Di dalam catur barat apabila bidak lawan tertangkap maka bidak tersebut keluar dari permainan sehingga seiring berjalannya waktu, jumlah bidak di papan permainan akan terus berkurang. Namun dalam permainan shogi, bidak yang tertangkap berpindah pihak sehingga selain menggerakan bidang di dalam papan permainan, pemain juga memiliki pilihan untuk meletakkan bidak yang telah ditangkap sebagai salah satu bidaknya.

C. Akhir dari Permainan

Pada permainan shogi, terdapat beberapa kondisi yang menyebabkan permainan berakhir. Kondisi-kondisi tersebut antara lain:

1) Resignation

Dalam permainan shogi, checkmate jarang terjadi. Hal ini disebabkan karena biasanya pemain menyerah atau mengundurkan diri setelah merasa bahwa tidak dapat menghindari kekalahannya. Dimana pengunduran diri biasanya dilakukan dengan cara membungkuk sambil mengatakan “Aku Kalah” atau meletakkan tangan kanan di atas piece stand (tempat diletakaannya bidak lawan yang tertangkap). Di dalam budaya barat, digantikan oleh jabat tangan sebagai alternatif tanda pengunduran diri.

2) Checkmate

Checkmate ditandai dengan bidak raja telah terpojok dan tidak terdapat langkah apapun untuk melindunginya. Atau dapat diartikan juga bahwa pemain lawan tidak dapat bergerak lagi sehingga pemain harus menyerah atau mengundurkan diri. Walaupun dalam kenyataannya pemain yang memiliki situasi buruk biasanya mengundurkan diri terlebih dahulu sebelum checkmate terjadi.

3) Illegal Move

Dalam turname shogi baik turnamen amatir atau profesional, pemain yang melakukan langkah ilegal akan langsung dinyatakan kalah. Hal ini juga berlaku apabila permainan telah berlangsung dan ditemukan langkah ilegal pada langkah-langkah sebelumnya. Namun apabila pemain lawan atau pihak ketiga tidak menyadari terdapat langkah ilegal sehingga pada akhirnya lawan menyerah

maka permainan berakhir dengan kekalahan pemain lawan. Akan tetapi pada permainan umum, biasanya aturan ini lebih lenggang dan pemain yang melakukan langkah ilegal diijinkan untuk mengganti langkahnya.

4) Repetition

Jika sebuah posisi yang sama dalam permainan terjadi 4 kali dengan giliran pemain yang sama, maka salah satu pemain akan dinyatakan kalah apabila semua langkah miliknya dalam pengulangan ini berakhir dalam kondisi check, apabila tidak berakhir check maka permainan dinyatakan seri. Akan tetapi hal ini sangatlah jarang terjadi mengingat dalam 4 kali, dimana setiap iterasinya harus memiliki posisi yang benar-benar sama, termasuk semua bidak di tangan (bidak di papan dan bidak yang tertangkap) harus sama persis.

5) Impasse/Deadlock

Kondisi Impasse terjadi apabila kedua bidak raja telah mencapai zona promosi milik pemain masing-masing serta tidak ada pemain yang dapat melakukan checkmate atau memperoleh bidak lagi. Kondisi ini dapat berakhir dengan kemenangan salah satu pihak atau seri. Ketika kedua pihak setuju terjadinya impasse maka dilakukan perhitungan poin: setiap bidak rook dan bishop yang dimiliki pemain bernilai 5, sedangkan bidak lain bernilai 1. Apabila pemain memiliki total poin kurang dari 24 maka akan dinyatakan kalah, namun apabila kedua pemain sama-sama memiliki poin kurang dari 24 atau lebih dari 24 maka permainan dinyatakan seri.

III. MONTE-CARLO TREE SEARCH [4][5][6][7]

Pada sub bab ini, penulis akan menjelaskan secara singkat terkait Monte-Carlo Tree Search (MCTS), algoritma dasar MCTS, Karakteristik MCTS, serta kelebihan dan kelemahan dari algoritma MCTS.

A. Pengenalan MCTS

Monte-Carlo Tree Search adalah salah satu metode pengambilan keputusan yang paling optimal dalam ruang lingkup permasalahan yang diberikan, dimana cara pengambilan sample dilakukan secara acak dan dibangunlah sebuah pohon pencarian sesuai dengan hasil-hasil yang diperoleh. MCTS sendiri memiliki daya tarik tersendiri yaitu merupakan bagian dari statistical anytime algorithm, dimana kapanpun algoritma ini dihentikan maka tetap dapat memberikan solusi akan tetapi tingkat kebaikan dari solusi tersebut bergantung kepada seberapa lama atau seberapa besar komputasi dilakukan.

Untuk setiap iterasi, sebuah tree policy digunakan untuk menemukan node yang paling penting dalah sebuah tree. Tree policy akan berusaha untuk menyeimbangkan antara eksplorasi (pencarian pada area yang belum diambil) dan exploitasi (pencarian pada area yang menjanjikan). Sebuah simulasi kemudian akan dijalankan dari node yang dipilih dan tree akan diperbaharui sesuai dengan hasil yang didapat. Penentuan langkah yang dibuat dalam proses simulasi ditentukan oleh default policy dimana default policy paling sederhana adalah langkah acak (random movement).

B. Algoritma Dasar MCTS

Algoritma dasar dari MCTS melibatkan pembangunan pohon pencarian secara iterative hingga waktu, memori, atau batasan iterasi tercapai. Ketika batasan tercapai, maka pencarian dihentikan dan langkah terbaik yang ditemukan akan dikembalikan sebagai hasil pencarian. Setiap node di pohon pencarian akan melambangkan sebuah state sedangkan hubungan menuju child node melambangkan langkah yang menghasilkan state baru pada child node tersebut. Terdapat empat tahapan pada setiap iterasi algoritma MCTS:

1) Selection

Dimulai dari root, sebuah policy akan digunakan untuk memilih child node dengan menelusuri pohon pencarian hingga ditemukan sebuah node yang perlu dikembangkan. Sebuah node dikatakan dapat dikembangkan apabila node tersebut tidak melambangkan terminal state dan masih memilih child node yang belum dikunjungi.

2) Expansion

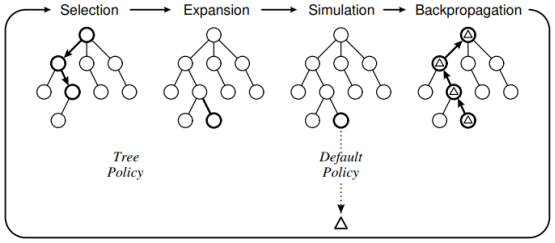
Satu atau beberapa child node akan ditambahkan untuk mengembangkan pohon pencarian berdasarkan langkah-langkah / rule permainan yang tersedia.

3) Simulation

Simulasi akan dijalankan dari node baru sesuai dengan default policy yang ada untuk memperoleh hasil akhir.

4) Backpropagation

Hasil dari simulation akan digunakan untuk memperhabarui statistik dari node-node pendahulunya.



Gambar. 2. Empat Tahap dalam 1 Iterasi MCTS

Dalam keempat tahap tersebut terdapat dua buah policy yang digunakan yaitu Tree Policy yang digunakan saat tahapan Selection dan Expansion, beserta Default Policy yang digunakaan saat tahapan Simulation dan Backpropagation. Dimana Tree Policy merupakan aturan untuk memilih atau menambahkan leaf node dari node-node yang telah ada di pohon pencarian sedangkan

Default Policy adalah proses menjalankan ruang lingkup permasalahan dari sebuah non terminal state untuk mendapatkan nilai perkiraan.

Ketika pencarian dihentikan (saat batasan tercapai) makan pencarian akan berhenti dan langkah terbaik akan dipilih berdasarkan 4 kriteria berikut ini:

1) *Max Child:* Memilih child node dari root dengan nilai hasil terbaik.

2) *Robust Child:* Memilih child node dari root yang paling sering dikunjungi.

3) *Max-Robust Child:* Memilih child node dari root yang memiliki hasil terbaik dan paling sering dikunjungi. Jika child node tidak ditemukan, maka lanjutkan pencarian hingga jumlah kunjungan yang cocok telah tercapai.

4) *Secure Child:* Memilih child node dari root yang memaksimalkan lower confidence bound.

C. Algoritma UCT

Tujuan algoritma MCTS adalah menentukan nilai permainan dari langkah yang dapat diambil dari state saat ini. Hal ini dicapai dengan membangun pohon pencarian dimana pembentukan pohon permainan tersebut bergantung pada cara pemilihan node di dalam pohon pencarian. Kesuksesan MCTS, terutama pada permainan bergantung pada Tree Policy yang digunakan. Secara khusus, *Kocsis dan Szepesvari*, mengajukan penggunaan UCB1 sebagai Tree Policy dimana pemilihan child node dianggap sebagai sebuah permasalahan multi armed bandit.

Multi armed bandit adalah permasalahan dimana seorang penjudi dihadapkan pada sebaris slot machine dan harus menentukan mesin manakah yang harus dimainkan, berapa kali harus memainkan setiap mesin, dan urutan untuk memainkan mesin-mesin tersebut. Ketika dimainkan, setiap mesin akan memberikan hadiah secara acak dari distribusi probabilitas yang khusus mesin tersebut. Tujuan dari penjudi adalah untuk memaksimalkan jumlah hadiah yang diperoleh melalui serangkaian penarikan tuas mesin.

UCB1 sangat sederhana dan efisien sehingga dijamin berada di dalam konstanta factor kemungkinan terbaik dan merupakan kandidat menjajikan untuk mengatasi dilemma explaration dan exploitation di MCTS. Setiap sebuah node akan dipilih dari pohon pencarian, dapat dimodelkan sebagai permasalahan multi armed bandit yang independen. Sehingga dapat dikatakan UCT adalah UCB1 yang diterapkan pada pohon pencarian atau yang bisa disebut Upper Confidence Bound applied to Tree.

$$UCT = \overline{X}_j + C_p\sqrt{\frac{2\ln n}{n_j}} \quad (1)$$

Dimana n adalah berapa kali node tersebut telah dikunjungi, n_j adalah berapa kali child node j telah dikunjungi dan C_p adalah sebuah konstanta yang bernilai 1.4. Sementara X_j merupakan tingkat kemenangan yaitu pembagian antara jumlah kemenangan dalam simulasi dengan jumlah kunjungan terhadap node tersebut. Apabila $n_j = 0$ maka dapat dipastikan bahwa UCT akan bernilai tidak terhingga, sehingga child node yang belum dikunjungi sebelumnya akan diberi nilai sebesar mungkin dengan tujuan untuk memastikan bahwa semua child akan ikut dipertimbangkan setidaknya satu kali sebelum salah satu child node dikembangkan lebih jauh.

D. Karakteristik MCTS

MCTS memiliki beberapa karakteristik yang membuatnya menjadi salah satu algoritma popular untuk berbagai macam permasalahan, antara lain:

1) Aheuristic

Tidak diperlukan pengetahuan khusus terhadap ruang lingkup permasalahan sehingga MCTS dapat digunakan pada permasalahan apapun yang dapat dimodelkan menggunakan pohon pencarian.

2) Anytime

MCTS Mengembalikan nilai dengan backpropagation secara langsung sehingga dapat dipastikan bahwa seluruh nilai di dalam pohon pencarian selalu merupakan nilai terbaru untuk setiap iterasinya. Sehingga MCTS dapat mengembalikan langkah dari state root kapanpun dibutuhkan.

3) Asymmetric

Selection pada MCTS melakukan pemilihan terhadap node-node yang lebih menjanjikan sehingga menyebabkan terbentuknya pohon pencarian yang tidak simetris seiring berjalannya waktu. Sehingga dapat dikatakan pembangunan pohon pencarian akan lebih condong ke area yang lebih menjanjikan.

IV. METODE DAN INTI PENELITIAN

Di dalam pembuatan kecerdasan buatan permainan shogi menggunakan algoritma MCTS, dibutuhkan beberapa hal yang harus menjadi perhatian khusus yaitu:

A. Elo Rating dan Bredley-Terry Model

Proses simulasi pada MCTS biasanya dilaukan denngan memilih secara acak hingga permainan berakhir, akan tetapi pada kebanyakan penerapan MCTS digunakan heuristik untuk melakukan pemilihan langkah yang lebih menjanjikan dibandingkan langkah lain. Akibatnya hasil simulasi menjadi lebih baik dibandingkan saat simulasi dilakukan dengan menggunakan langkah acak. Salah satu heuristik yang dapat digunakan pada permainan papan adalah Elo Rating. Dimana Elo Rating akan memberikan

nilai ketika sebuah langkah dipilih berdasarkan kombinasi fitur-fitur atau komponen-komponen permainan saat itu.

Selain menggunakan Elo Rating, metode yang digunakan untuk memperkirakan kemungkinan sebuah langkah dipilih didasarkan pada Bradley-Terry Model. Dimana model tersebut akan memprediksi hasil dari kompetisi baik secara individual ataupun group. Kemungkinan individu i memenangkan kompetisi dapat dirumuskan sebagai berikut:

$$P(iwins) = \frac{\gamma_i}{\sum_{j=1}^{n} \gamma_j} \quad (2)$$

Selain kompetisi antar individual, Bradley-Terry Model juga dapat memprediksi kompetisi antar tim/group. Dimana kekuatan dari group dapat diperkirakan dari kombinasi kekuatan masing-masing anggota group tersebut. Sebagai contoh, kemungkinan sebuah tim terdiri dari dua individu (individu 4 dan 5) untuk memenangkan kompetisi terhadap tim dengan tiga individu (individu 1-3) atau tim dengan individu tunggal (individu 6) dapat dihitung dengan menggunakan rumus:

$$P(4-5wins) = \frac{\gamma_4 \gamma_5}{\gamma_1 \gamma_2 \gamma_3 + \gamma_4 \gamma_4 + \gamma_6} \quad (3)$$

Dalam permainan shogi sendiri sebuah langkah dapat memiliki beberapa fitur perhitungan. Oleh karena itu dalam Breadley Terry Model, fitur perhitungan merupakan individu, sementara langkah bidak merupakan sebuah tim atau group. Kemungkinan sebuah langkah dipilih berhubungan denngan kemungkinan menang dari sebuah tim yang berisikan individu fitur-fitur perhitungan dari langkah tersebut.

B. Fitur-fitur Perhitungan [8]

Terdapat 7 fitur perhitungan yang digunakan untuk menghitung kemungkinan sebuah langkah memenangkan kompetisi dibandingkan langkah lainnya. Berdasarkan rujukan elo rating untuk permainan GO, Catur, dan shogi, berikut fitur beserta nilai elo rating yang digunakan dalam penelitian ini, antara lain:

TABEL I
FITUR DAN NILAI ELO RATING YANG DIGUNAKAN

Fitur Perhitungan	Detail Fitur	γ_i
Material Balance (SEE)	> 650	20.4
	550 – 650	14.37
	450 – 550	9.54
	350 – 450	6.08
	250 – 350	3.89
	150 – 250	2.90
	50 – 150	2.55
	(-50) – 50	1.11
	(-150) – (-50)	0.47
	(-250) – (-150)	0.35
	(-350) – (-250)	0.32
	(-450) – (-350)	0.10
	(-550) – (-450)	0.06
	(-650) – (-550)	0.05
	< (-650)	0.02
Check		7.55
Promote		1.47
Escape	Pawn	1.13 – 1.27
	Lance	1.90 – 3.19
	Knight	1.70 – 2.59
	Silver	2.20 – 2.60
	Gold	4.38 – 4.78
	Bishop	1.77 – 5.12
	Rook	6.85 – 23.68
	Promoted Bishop	3.26 – 9.98
	Promted Rook	8.53 – 14.62
	Promoted Piece Lain	0.82 – 1.65
Capture	Recapture	12.75
	Capture Biasa	1.88
King Danger	Tidak Kondisi Check	0.30 – 1.01
	Kondisi Check	0.57 – 4.21
Piece Drop Value	Pawn	0.50 – 2.21
	Knight	0.24 – 0.97
	Lance	0.17 – 1.66
	Silver	0.19 – 1.35
	Gold	0.28 – 1.45
	Bishop	0.16 – 1.17
	Rook	0.22 – 1.89

Nilai Elo Rating pada tabel I inilah yang nantinya akan digunakan oleh Bradley-Terry Model untuk menghitung kemungkinan sebuah langkah dipilih. Berikut ini adalah penjelasan singkat terkait masing-masing fitur:

1) Material Balance (SEE)

Dalam menghitung Material Balance digunakan SEE (Static Exchange Evaluation) dimana SEE memeriksa konsekuensi dari hasil pertukaran yang terjadi pada sebuah kotak dalam papan permainan setelah sebuah langkah dilakukan dan menghitung perubahan evaluasi yang diperoleh atau hilang. Nilai evaluasi positif melambangkan bahwa pertukaran ini baik sementara nilai negatif melambangkan pertukaran itu buruk.

歩 (Pawn)	香 (Lance)	桂 (Knight)	銀 (Silver)	金 (Gold)	角 (Bishop)	飛 (Rook)
100	280	300	420	530	620	700
と (+Pawn)	成香 (+Lance)	成桂 (+Knight)	成銀 (+Silver)	-	馬 (+Bishop)	龍 (+Rook)
270	320	250	430	-	710	850

Gambar. 3. Nilai dari Setiap Bidak (+ adalah bidak promosi)

2) Check

Fitur check melihat apakah sebuah langkah akan memberikan check kepada raja milik lawan atau tidak.

Jika sebuah langkah memberikan check bidak raja maka langkah tersebut adalah langkah yang baik.

3) Promotion

Berbeda dengan catur, hampir seluruh bidak di dalam shogi dapat dipromosikan dimana bidak yang telah mengalami promosi lebih berharga daripada bidak yang tidak promosi hal ini disebabkan promosi menyebabkan bidak berubah langkah di dalam shogi.

4) Escape

Escape merupakan fitur yang dilihat saat bidak berada dalam area serang lawan dimana apabila bidak diserang oleh lawan yang memiliki nilai lebih kecil maka merupakan langkah yang baik ketika bidak melarikan diri.

5) Capture dan Recapture

Penangkapan yang baik atau buruk harus selalu diperhatikan pada permainan shogi mengingat pada permainan ini bidak yang ditangkap akan berganti pihak. Oleh sebab itu, bidak yang terlihat kurang berharga sekalipun dapat digunakan untuk menangkap raja. Perbedaan dari capture dan recapture adalah ketika jumlah bidak kita belum berkurang maka disebut capture sedangkan ketika jumlah bidak kita sudah berkurang maka disebut juga dengan recapture.

6) King Danger

Dikatakan raja dalam bahaya adalah ketika banyaknya jumlah serangan pada kotak di samping raja. Sehingga langkah yang meningkatkan hal ini di sisi kita adalah langkah yang buruk. Akan tetapi merupakan langkah yang baik ketika langkah yang dipilih adalah langkah yang mengurangi hal tersebut.

7) Piece Drop Value

Fitur ini adalah fitur yang diberikan ketika sebuah bidak berasal dari reserve diletakan ke papan permainan. Nilai Elo Rating yang digunakan adalah nilai yang sama yang digunakan pada fitur material balance. Akan tetapi kita harus berhati-hati akan fitur ini, agar bidak yang baru diletakan tidak dimakan pada giliran berikutnya.

C. Progressive Widening [9]

Selain menggunakan Elo Rating, di penelitian ini progressive widening juga ditambahkan pada UCT berdasarkan rumus berikut ini:

$$U_i = \begin{cases} dR_i + c'\sqrt{\frac{2\log n}{e}} & (n_i = 0) \\ X_i + c\sqrt{\frac{2\log n}{n_i}} & (n_i \neq 0) \end{cases} \qquad (4)$$

$$I = \underset{i \in \{1,\dots,K\}}{argmax} \{U_i\}$$

Konstanta c', d, dan e masing-masing memiliki nilai 0.5, 4.0, dan 10.0. Sementara nilai c adalah 1.4 dan R_j adalah nilai Elo Rating dari sebuah langkah permainan yang dinormalisasi menggunakan rumus 5.

$$R_i = \frac{\gamma_i}{\underset{j \in \{1,\dots,K\}}{argmax}\{\gamma_j\}} \qquad (5)$$

Dengan menggunakan Progressive Widening, maka UCT lebih cenderung mengembangkan langkah dengan nilai Elo Rating tinggi seperti Recapture atau Check. Selain itu, dengan penambahan progressice widening apabila sebuah node belum dikunjungi maka nilai yang dihasilkan bukan bernilai tidak berhingga mengingat root node setidaknya telah dikunjungi sekali.

D. Killer Move dan History Heuristic [9]

Selain penambahan Progressive Widening, pada penelitian ini juga ditambahkan metode Killer Move dan History Heuristic untuk rumus UCT. Kedua metode ini digunakan untuk mengatur nilai konstanta c di rumus UCT (Rumus 4). Killer move adalah node terbaik pada jajaran sibling atau dapat dikatakan juga node terbaik pada child node yang sedang dilakukan ekspansi. Pemilihan node terbaik ini didasarkan pada frekuensi kunjungan (Robust Child). History Heuristic adalah rasio sebuah node menjadi node terbaik dalam proses pencarian. Penentuan rasio tersebut menggunakan rumus rating node dibagi total rating semua sibling yang ada.

V. HASIL UJI COBA

Ada beberapa ujicoba yang dilakukan pada penelitian ini yaitu ujicoba melawan AI Program Lain, uji coba melawan Algoritma Negamax Alpha Beta Prunning, dan uji coba User Acceptance. Berikut adalah penjelasan lebih detail terkait hasi masing-masing uji coba.

A. Uji Coba Melawan AI Program Lain

Uji coba ini menandingkan MCTS yang telah dikembangkan di penelitian ini melawan AI 2 program shogi yang telah direlease di masyarakat saat ini. Masing-masing program memiliki beberapa tingkat kesulitan dalam bermain shogi. Uji coba ini dilakukan dengan memberi waktu 10 detik (pemilihan 10 detik didasarkan pada rata-rata waktu pergerakan AI pada umumnya) kepada MCTS untuk berpikir sebelum mengembalikan langkah yang dipilih. Selain itu ujicoba dilakukan 5x untuk masing-masing tingkat kesulitan program tandingan.

TABEL III
HASIL UJI COBA MELAWAN AI PROGRAM LAIN

Program	Tingkat Kemenangan		
	Beginner	Intermediate	Hard
1	100%	60%	20%

2	100%	40%	0%

Dapat dilihat pada tabel II bahwa 10 detik MCTS mampu dengan mudah mengalahkan tingkat kesulitan beginner pada 2 program tandingan akan tetapi kesulitan dalam menghadapi Tingkat kesulitan intermediate dan hard. Hal ini dikarenakan pada tingkat intermediate dan hard, pada umumnya pengembang game shogi menggunakan formasi sehingga sering membuat MCTS salah langkah. Apabila kita meningkatkan waktu pencarian solusi MCTS, maka dapat dipastikan tingkat kemenangan juga akan meningkat akan tetapi apabila waktu pencarian terlalu lama, maka dapat menyebabkan user yang memainkannya menjadi kurang tertarik.

B. Uji Coba Melawan Algoritma Negamax

Uji coba ini menandingkan MCTS dengan algoritma Negamax Alpha Beta Prunning, dimana uji coba dilakukan dengan memberikan waktu 10 detik kepada MCTS sebelum mengembalikan langkah yang dipilih. Algoritma Negamax Alpha Beta Prunning menggunakan depth 3 dan pertandingan dilakukan 20x.

TABEL IIIII
HASIL UJI COBA MELAWAN NEGAMAX

	MCTS	**Negamax Alpha Beta Prunning**
Tingkat Kemenangan	80%	20%

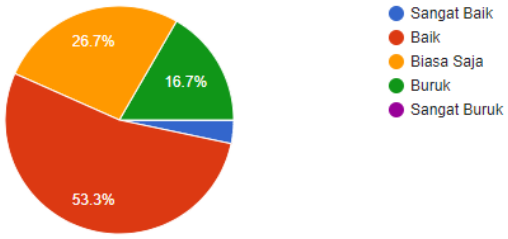
Pada tabel III, dapat dilihat bahwa MCTS memenangkan pertandingan sebanyak 16x dari 20 pertandingan. Mengingat depth pada Negamax Alpha Beta Prunning adalah 3 maka hasil yang dicapai cukup baik. Apabila depth dari negamax ditingkatkan diatas 3, maka waktu yang dibutuhkan oleh negamax untuk berpikir menjadi sangat lama dikarenakan berbeda dengan MCTS yang dapat dengan mudah dihentikan oleh waktu, negamax mewajibkan pencarian di seluruh node yang ada.

C. Uji Coba User Acceptance

Walaupun fokus pada penelitian ini adalah MCTS pada pernainan Shogi tetapi tetap dibutuhkan user acceptance untuk menentukan apakah aplikasi hasil penelitian ini ataupun kecerdasan yang dihasilkan memenuhi standar dari permainan shogi atau tidak. Uji coba dilakukan oleh 30 pemain shogi dimana masing-masing pemain memberikan penilaian pada kuesioner yang dibagikan. Berikut adalah 2 pertanyaan penting di kuesioner:

1) UI pada Permainan apakah User Friendly

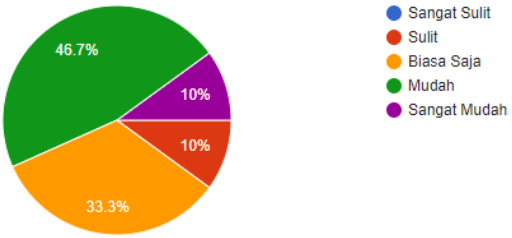
Untuk menentukan apakah sebuah aplikasi layak pakai atau tidak, UI pada aplikasi tersebut wajib user friendly. Gambar 4 merupakan grafik hasil kuesiorer dimana 53.3% menyatakan bahwa UI aplikasi baik tetapi sangat disayangkan 16.7% mengatakan buruk dan kurang user friendly.



Gambar. 4. Grafik Kuesioner UI Aplikasi

2) Tingkat Kesulitan AI Shogi

Dikarenakan fokus penelitian ini adalah AI pada permainan shogi maka salah satu pertanyaan yang diberikan terkait Tingkat Kesulitan AI bagi pemain shogi. Gambar 5 merupakan grafik kuesioner tingkat kesulitan AI, dimana 10% menyatakan sulit tetapi mayoritas menyatakan mudah (46.7%) atau bahkan menyatakan sangat mudah (10%).



Gambar. 5. Grafik Kuesioner Tingkat Kesulitan AI

VI. KESIMPULAN

Berdasarkan ujicoba yang telah dilakukan pada penelitian ini, berikut beberapa kesimpulan yang diambil antara lain:

1. MCTS dengan waktu perhitungan 10 detik dapat disetarakan dengan tingkat beginner hingga intermediate ketika dibandingkan dengan AI Program Shogi Lain.
2. Dibandingkan algoritma Negamax Alpha Beta Pruning, MCTS memiliki kelebihan dalam menentukan waktu pengembalian hasil dimana Negamax tidak dapat diatur waktunya.
3. Selain penentuan waktu, penggunaan jumlah resource MCTS berjalan lebih minimal dibandingkan Negamax Alpha Beta Pruning dikarenakan pada Negamax resource meningkat seiring kedalaman yang digunakan dan jumlah piece yang harus digerakan atau dimainkan.
4. Berdasarkan user acceptance, tingkat kesulitan AI 10 detik MCTS dirasa terlalu mudah bagi mayoritas pemain shogi.

DAFTAR PUSTAKA

[1] J. Fairbairn. "Shogi for Beginners". Kiseido Publishing. 1984.
[2] T. Leggett. "Japanese Chess". Tuttle Publishing. 2009.
[3] Y. Habu and T. Hosking. "Masters of Shogi: Games Collection". 2010.
[4] C. Browne, S. Colton, P. I. Cowling, S. Lucas, D. Perez, E. Powley, P. Rohlfshagen, S. Samothrakis, S. Tavener, D. Whitehouse. "A Survey of Monte Carlo Tree Search Method". 2012.
[5] R. Coulom. "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search". 2008.
[6] L. Kocsis, C. Szepesvari. "Bandit-Based Monte-Carlo Planning". 2006.
[7] R. J. Lorentz. "Amazons Discover Monte-Carlo". 2008.
[8] R. Coulom. "Computing Elo Ratings of Move Patterns in the Game of Go". 2007.
[9] S. Gelly. R. Munos. O. Teytaud. W. Yizao. "Modifications of UCT Patterns in Monte-Carlo Go". 2006.