

Penerapan Alpha-Beta Pruning dalam Pengembangan AI Catur: Evaluasi dan Peningkatan Strategi

Nandana Wahyu R¹, Awang Mohammad Ziadhasya Rizqaarafi²,
Moh Quthbul Widad³, Anggraini Puspita Sari⁴

^{1,2,3,4}Universitas Pembangunan Nasional “Veteran” Jawa timur, Kota Surabaya

Email :

¹22081010298@student.upnjatim.ac.id, ²22081010327@student.upnjatim.ac.id,

³22081010319@student.upnjatim.ac.id,, ⁴anggraini.puspita.if@upnjatim.ac.id

Abstrak

Permainan catur merupakan permainan strategis yang dimainkan oleh dua pemain untuk memetakan pergerakan lawan mereka. Ketika bermain catur, orang belajar berpikir abstrak dan mencoba untuk menyelesaikan masalah, hal ini dapat berpengaruh terhadap pola pikir seseorang dalam kehidupan sehari-hari. Seiring berkembangnya zaman, teknologi digital mengalami bertumbuh pesat dan berdampak pada sistematis permainan catur itu sendiri. Penerapan algoritma alpha-beta pruning dalam pengembangan kecerdasan buatan untuk permainan catur telah menunjukkan peningkatan efisiensi dalam proses pencarian langkah yang optimal. Penelitian ini bertujuan untuk mengevaluasi pengaruh tingkat kedalaman alpha-beta pruning terhadap kesulitan dan waktu komputasi dalam permainan catur. Algoritma ini diimplementasikan menggunakan bahasa pemrograman Python di lingkungan pengembangan Visual Studio Code (VS Code) dengan pustaka Pygame untuk antarmuka grafis. Algoritma alpha-beta pruning mampu mengoptimalkan pencarian dengan memangkas cabang yang tidak relevan, namun membutuhkan sumber daya komputasi yang lebih besar pada kedalaman yang lebih tinggi.

Kata Kunci: Alpha-beta pruning, kecerdasan buatan, catur, algoritma pencarian, optimasi.

Abstract

Chess is a strategic game played by two players with the goal of checkmating their opponent. When playing chess, people learn to think abstractly and solve problems, which can influence their thought patterns in daily life. As time progresses, digital technology has grown rapidly, impacting the systematic aspects of the game of chess itself. The application of the Alpha-Beta Pruning algorithm in the development of artificial intelligence (AI) for chess has demonstrated increased efficiency in the process of searching for optimal moves. This study aims to evaluate the impact of Alpha-Beta Pruning depth levels on difficulty and computation time in chess. This algorithm is implemented using the Python programming language in the Visual Studio Code (VS Code) development environment with the Pygame library for the graphical interface. The alpha-beta pruning algorithm can optimize the search by pruning irrelevant branches, but it requires greater computational resources at higher depths.

Keywords: *Alpha-beta pruning, artificial intelligence, chess, search algorithm, optimization.*

PENDAHULUAN

Di era ini, masyarakat sangat suka untuk memainkan sebuah permainan karena adanya sebuah target yang harus dicapai saat memainkan permainan (Utami, 2013). Individu cenderung mengamati, berpikir, memilih, dan berinovasi untuk mencapai tujuan tersebut (Brooks, 2022). Catur berfungsi sebagai media untuk mengasah keterampilan-keterampilan tersebut. Catur merupakan permainan strategi yang dimainkan oleh dua pemain dan mengharuskan mereka untuk mematikan gerak lawan mereka, telah lama dianggap sebagai alat yang sangat baik untuk mengasah kemampuan kognitif. Kemampuan kognitif seperti ingatan, perhatian, dan kemampuan berpikir kritis dilatih dalam permainan catur, yang menuntut pemikiran logis, analitik, dan strategis (Nanu et al, 2023). Ketika bermain catur, orang belajar berpikir abstrak dan menyelesaikan masalah, yang penting untuk berbagai aspek kehidupan, seperti pendidikan, karir, dan kehidupan sehari-hari.

Beberapa tahun belakangan ini, teknologi digital bertumbuh sangat pesat dan berdampak pada peningkatan pemanfaatan teknologi yang terus meningkat (Sari et al., 2022). Salah satunya adalah kecerdasan buatan. Kecerdasan buatan atau artificial intelligence sudah banyak digunakan baik dalam aplikasi penunjang sehari-hari seperti *Google Maps*, *Google Assistant*, *OpenAI*, dan masih banyak lagi. Tidak hanya itu, dalam aplikasi permainan komputer sekarang juga sudah menggunakan kecerdasan buatan untuk meningkatkan realisme, mengembangkan strategi permainan yang lebih kompleks, dan memberikan pengalaman bermain yang lebih interaktif dan menantang bagi pengguna. Sekarang pun sudah banyak orang yang berlomba-lomba untuk mengembangkan permainan komputer yang berbasis kecerdasan buatan. Contohnya adalah permainan catur.

Permainan ini menjadi medan yang menarik bagi penelitian dalam bidang kecerdasan buatan karena kompleksitasnya dan aturan yang jelas. Permainan catur terdiri dari tiga fase: pembukaan, permainan tengah, dan akhir permainan. Setiap fase memiliki strategi dan taktik khusus untuk mendapatkan keunggulan dan mencapai skakmat. Pembukaan fokus pada pengembangan buah catur dan kontrol pusat papan, permainan tengah fokus pada manuver taktis dan perhitungan, sementara akhir permainan fokus pada pemanfaatan buah catur yang tersisa untuk skakmat (Root, 2008; Trincherro & Sala, 2016).

Membuat komputer untuk bermain permainan papan sudah lama menjadi fokus para ilmuwan sejak komputer dibuat. Komputer pertama yang berhasil mengalahkan manusia dalam permainan catur terbuat pada tahun 1956, komputer sudah dapat mengalahkan manusia dengan kemampuan rendah-menengah (Bill Wall, 2012). Kecerdasan buatan sudah digunakan dalam permainan komputer sejak awal kemunculannya. Sebuah program kecerdasan buatan dapat menjadi lawan yang sepadan bahkan sulit untuk dikalahkan untuk lawannya (Ribeiro et al., 2017). Alpha-beta pruning merupakan salah satu algoritma yang dapat digunakan sebagai mesin kecerdasan permainan catur.

Latar belakang pengimplementasian alpha-beta pruning pada catur (atau permainan catur) terletak pada kebutuhan untuk meningkatkan efisiensi dalam pencarian permainan yang memiliki jumlah pergerakan yang sangat besar. Dalam permainan catur, setiap permainan memiliki potensial sekitar 10^{120} pergerakan, yang membuatnya tidak praktis untuk mencoba semua pergerakan yang mungkin dalam waktu yang wajar. Oleh karena itu, algoritma pencarian seperti alpha-beta pruning menjadi penting untuk memungkinkan mesin untuk mencari pergerakan terbaik dengan cara yang efisien. Penerapan alpha-beta pruning pada catur tidak hanya membantu dalam meningkatkan efisiensi

dalam pencarian pergerakan terbaik, tetapi juga memungkinkan mesin untuk membuat keputusan lebih cepat dalam permainan dengan ruang keadaan yang besar. Hal ini sangat penting dalam permainan catur, di mana kecepatan dan keakuratan dalam memilih pergerakan dapat menentukan kemenangan atau kekalahan.

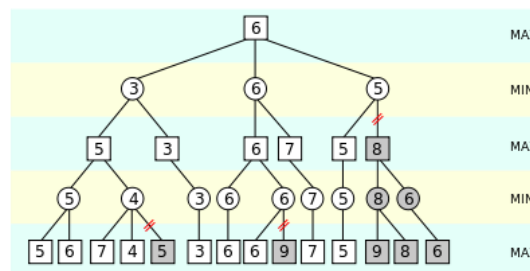
Selain itu, alpha-beta pruning juga memiliki potensi untuk memotong seluruh subpohon serta simpul individu, memberikan fleksibilitas dalam proses pencarian yang lebih efisien. Ini memungkinkan algoritma untuk menghentikan eksplorasi lebih lanjut pada cabang yang sudah ditemukan tidak menjanjikan, sehingga menghemat waktu dan sumber daya komputasi. Secara keseluruhan, pengimplementasian alpha-beta pruning pada catur menunjukkan bagaimana teknologi dapat digunakan untuk meningkatkan kemampuan mesin dalam memahami dan memprediksi permainan strategis seperti catur. Dengan mengurangi jumlah pergerakan yang perlu dievaluasi, algoritma ini memungkinkan mesin untuk menjadi pemain yang lebih cerdas dan efisien dalam permainan catur.

Penelitian ini bertujuan untuk mengetahui bagaimana tingkat kedalaman Alpha-Beta Pruning mempengaruhi kesulitan dalam permainan catur. Penelitian ini berfokus pada bagaimana algoritma alpha-beta pruning dapat mempengaruhi waktu dan kompleksitas permainan catur.

METODE

Catur

Penelitian ini menerapkan algoritma alpha-beta pruning untuk meningkatkan efisiensi pencarian langkah optimal dalam permainan catur. Pada gambar 1. Alpha beta pruning merupakan contoh pohon pencarian yang menggunakan algoritma alpha beta pruning.



Gambar 1. Alpha-beta pruning

Adapun penjelasan dari Gambar 1. Alpha-beta pruning sebagai berikut:

- a. **Minimax**
MIN dan MAX diasumsikan sebagai 2 pemain yang terlibat. Pemain maximal (Max) yang berusaha memaksimalkan skor dan pemain min (Min) yang berusaha meminimalkan skor.
- b. **Algoritma alpha beta pruning**
Alpha-Beta pruning adalah sebuah algoritma pencarian yang bertujuan untuk mengurangi jumlah nodus yang dievaluasi oleh algoritma minimax dalam pohon pencariannya (Singhal & Sridevi, 2019). Alpha, merupakan kondisi nilai terbaik yang bisa diambil oleh nilai Max. Sedangkan Beta, memaksimalkan skor yang dapat dicapai oleh pemain Min. Dalam permainan catur, alpha-beta pruning digunakan untuk mengurangi jumlah posisi yang perlu dievaluasi. Berikut adalah gambaran algoritma

```
def maxValue(self, state, alpha, beta, d):
    if time.time() - self.start_time > self.max_time:
        return self.eval(state), None
    if d == 0 or state.checkmate or state.stalemate:
        return self.eval(state), None

    possibleMoves = state.getValidMoves()
    if not possibleMoves:
        return self.eval(state), None

    value = -1000000
```

```

bestMove = None

for m in possibleMoves:
    state.makeMove(m)
    v2, _ = self.minValue(state, alpha,
beta, d - 1)
    state.undoMove()
    if v2 > value:
        value = v2
        bestMove = deepcopy(m)
    if value >= beta:
        return value, bestMove
    alpha = max(alpha, value)

return value, bestMove

def minValue(self, state, alpha, beta, d):
    if time.time() - self.start_time >
self.max_time:
        return self.eval(state), None
    if d == 0 or state.checkmate or
state.stalemate:
        return self.eval(state), None

    possibleMoves = state.getValidMoves()
    if not possibleMoves:
        return self.eval(state), None

    value = 10000000
    bestMove = None

    for m in possibleMoves:
        state.makeMove(m)
        v2, _ = self.maxValue(state, alpha,
beta, d - 1)
        state.undoMove()
        if v2 < value:
            value = v2
            bestMove = deepcopy(m)
        if value <= alpha:
            return value, bestMove
        beta = min(beta, value)

    return value, bestMove

```

‘Value’ pada algoritma ini akan mengembalikan nilai balik dari masing-masing Min dan Max. Pemain Max akan mencoba memilih gerakan dengan nilai maksimum. Sedangkan pemain Min akan mencoba meminimalisir keluaran dari pemain Max.

Catur

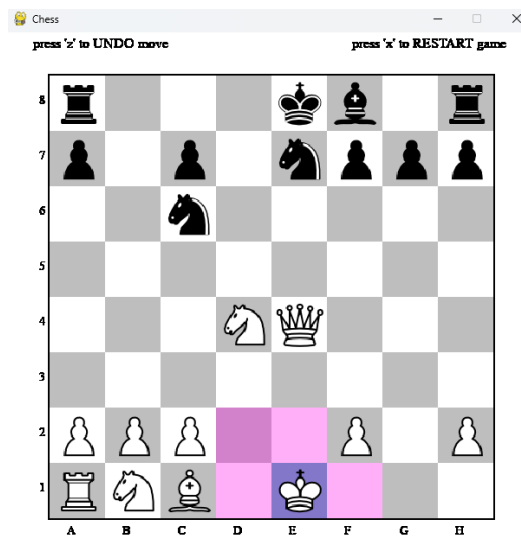
Permainan catur adalah permainan strategi dua pemain yang dimainkan di atas papan persegi dengan ukuran 8x8 yang terdiri dari 64 kotak yang berselang-seling antara warna terang dan gelap. Setiap pemain memulai permainan dengan 16 buah catur yang terdiri dari:

a. Langkah Node Catur

Permainan catur dimainkan oleh dua pemain di atas papan 8x8 kotak dengan 64 petak yang berselang-seling warna terang dan gelap. Setiap pemain memulai dengan 16 buah catur yang terdiri dari satu raja, satu ratu, dua benteng, dua kuda, dua gajah, dan delapan pion. Tujuan utama catur adalah skakmat, yaitu ketika raja lawan dalam ancaman serangan dan tidak bisa lolos dari ancaman tersebut (FIDE, 2020).

Gerakan Buah Catur:

- Raja: Bergerak satu petak ke segala arah. Contoh bisa dilihat dari Gambar 2. Pergerakan Raja



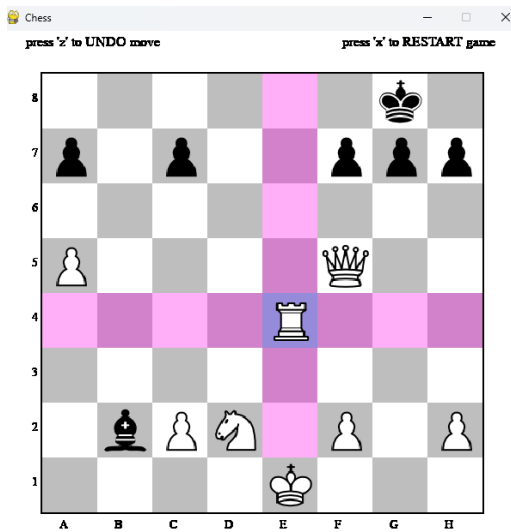
Gambar 2. Pergerakan Raja

- Ratu: Bergerak sejauh mungkin secara horizontal, vertikal, atau diagonal. Bisa dilihat dari Gambar 3. Pergerakan Ratu



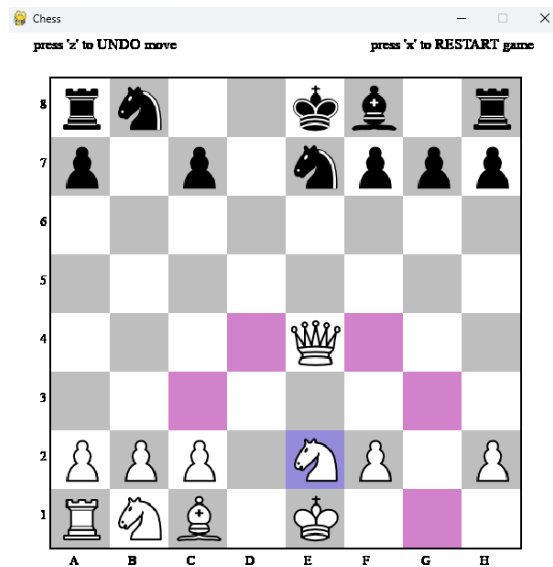
Gambar 3. Pergerakan Ratu

- Benteng: Bergerak sejauh mungkin secara horizontal atau vertikal. Contoh dapat dilihat pada Gambar 4. Gambar Benteng



Gambar 4. Gambar Benteng

- Kuda: Bergerak dalam pola 'L', dua petak dalam satu arah dan kemudian satu petak tegak lurus ke arah pertama. Contoh pada Gambar 5. Pergerakan Kuda



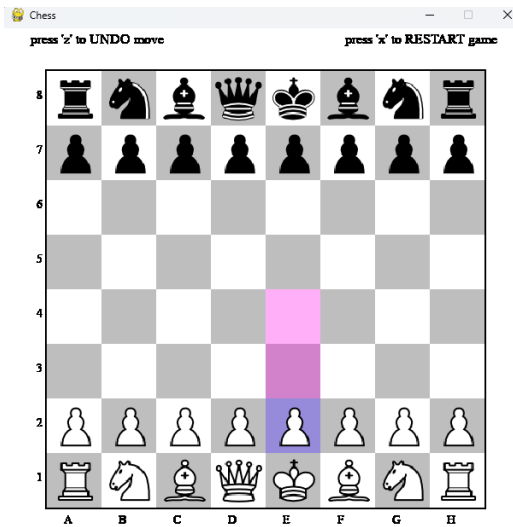
Gambar 5. Pergerakan Kuda

- Gajah: Bergerak sejauh mungkin secara diagonal. Contoh dapat dilihat dari Gambar 6. Pergerakan Kuda

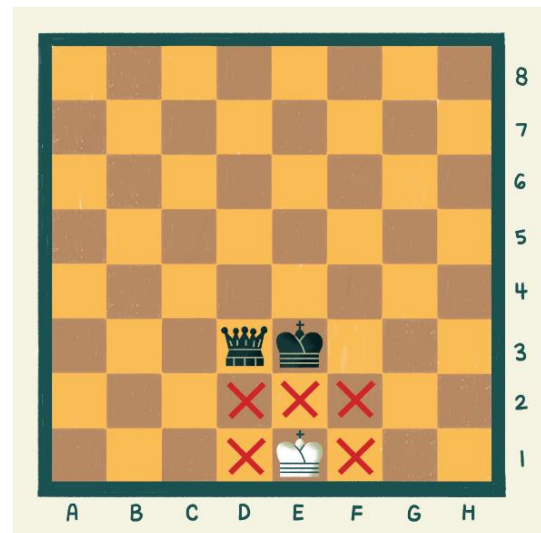


Gambar 6. Pergerakan Pion

- Pion: Bergerak satu petak ke depan, tetapi menangkap satu petak secara diagonal ke depan. Pion memiliki gerakan khusus yang disebut "en passant" dan dapat dipromosikan ketika mencapai baris terjauh (FIDE, 2020). Contoh pergerakan pion pada Gambar 7. Pergerakan Pion



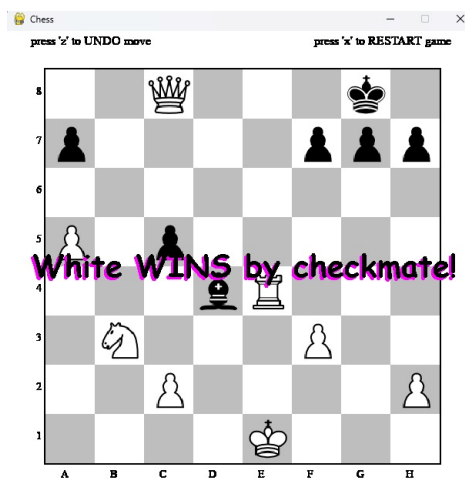
Gambar 7. Pergerakan Pion



Gambar 9. Stalemate

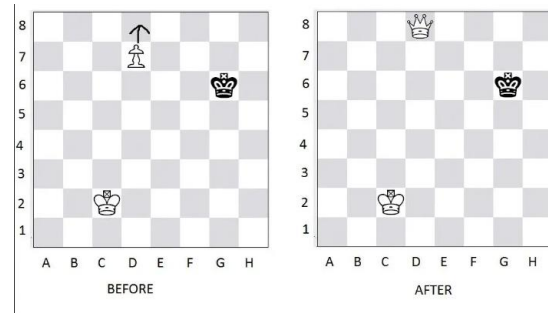
b. Kondisi Khusus:

- Skak: Ketika raja berada dalam ancaman serangan langsung tetapi raja masih bisa memiliki langkah untuk menyelematkan diri (Gobet & Campitelli, 2006).
- Skakmat: Ketika raja dalam skak dan tidak ada langkah legal yang bisa menghindari ancaman (Gobet & Campitelli, 2006). Contoh kondisi skakmat dapat dilihat pada gambar 8.



Gambar 8. Putih Checkmate

- Promosi Pion: Ketika pion mencapai baris terjauh dari posisinya, ia dapat dipromosikan menjadi ratu, benteng, gajah, atau kuda (FIDE, 2020). Contoh kondisi promosi pion dapat dilihat pada Gambar 10. Promosi Pion.



Gambar 10. Promosi Pion

- Patt (Stalemate): Ketika pemain tidak memiliki langkah legal yang bisa dilakukan dan rajanya tidak dalam skak, yang menghasilkan seri (Gobet & Campitelli, 2006). Contoh kondisi stalemate dapat dilihat pada Gambar 9. Stalemate.

- En Passant: Gerakan khusus di mana pion menangkap pion lawan yang telah bergerak dua kotak dari posisi awalnya (FIDE, 2020). Contoh dapat dilihat pada gambar 11.



Gambar 11. En Passant

Prosedur Pelaksanaan

1. Instalasi Alat dan Bahan:
 - Python: Versi 3.8 atau lebih baru.
 - Pygame: Instalasi dilakukan melalui terminal dengan perintah: “pip install pygame
2. Implementasi Algoritma:
 - Alpha-Beta Pruning: Algoritma alpha-beta pruning diterapkan pada pohon pencarian langkah-langkah permainan catur. Fungsi utama dari algoritma ini adalah mengurangi jumlah node yang dievaluasi selama pencarian, dengan memotong cabang yang tidak akan mempengaruhi keputusan akhir.
 - Representasi Papan Catur: Papan catur direpresentasikan sebagai array dua dimensi. Setiap posisi pada papan dapat diakses dan dimanipulasi oleh algoritma.
 - Evaluasi Posisi: Fungsi evaluasi digunakan untuk menilai kekuatan posisi di papan catur. Nilai numerik diberikan berdasarkan posisi dan jenis bidak, serta strategi permainan yang diimplementasikan.
3. Prosedur Eksperimen:
 - Inisialisasi Permainan: Permainan dimulai dengan menginisialisasi papan catur dan menempatkan bidak pada posisi awal.
 - Penerapan Alpha-Beta Pruning: Pada setiap giliran, algoritma

alpha-beta pruning digunakan untuk menentukan langkah terbaik yang tersedia. Kedalaman pencarian diatur untuk memastikan balasan yang efisien dalam waktu yang wajar.

- Antarmuka Pengguna: Antarmuka grafis dibangun menggunakan Pygame untuk menampilkan papan catur dan memungkinkan interaksi pengguna.

Penelitian ini merupakan jenis penelitian eksperimental, dimana akan dilakukan percobaan untuk melihat seberapa dalam dan efektif alpha-beta pruning dalam menjalankan permainan game catur.

Pengumpulan Data

Data-data yang dikumpulkan merupakan hasil permainan catur dengan kedalaman mencari yang berbeda. Hasil tersebut akan dijadikan bahan analisis untuk melihat kekurangan, kelebihan pada algoritma alpha-beta pruning.

HASIL DAN PEMBAHASAN

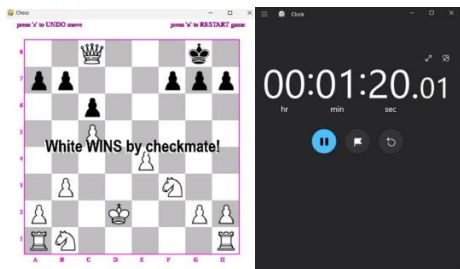
Hasil

Pada bab ini, akan dipaparkan hasil dari implementasi algoritma alpha-beta pruning dalam permainan catur. Algoritma ini merupakan optimisasi dari pencarian minimax yang bertujuan untuk mempercepat waktu komputasi dengan cara memangkas cabang-cabang pada pohon pencarian yang tidak perlu dievaluasi lebih lanjut. Parameter yang kami gunakan dalam penelitian ini yaitu kedalaman pencarian tree dan waktu yang diperlukan dalam setiap level permainannya. Berikut adalah tabel data sebagai berikut:

Tabel 1. Tabel Percobaan *Depth* (Kedalaman)

<i>Depth</i>	Waktu yang Diperlukan (Menit)
1	1,33
2	14,22
3	15,67

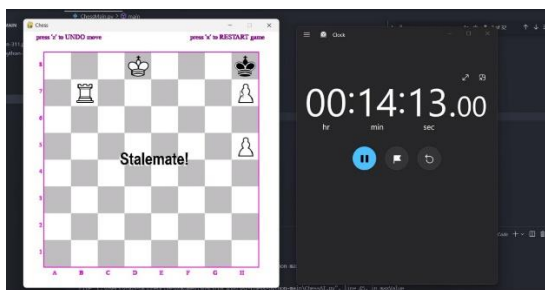
- *Depth* bermain 1



Gambar 12. *Depth* Bermain 1

Pada *depth* bermain 1, pencarian mesin catur akan berhenti pada kedalaman 1. Artinya mesin kecerdasan buatan catur tidak akan mencari kemungkinan-kemungkinan yang bisa terjadi lebih lanjut. Hal ini juga mengakibatkan singkatnya waktu komputasi mesin catur dalam melakukan langkah selanjutnya. Pada Gambar 12. *Depth* Bermain 1, terdapat kondisi kemenangan catur putih atau pemain manusia dinyatakan sebagai *checkmate*, dengan waktu 1.33 menit. Sehingga bisa kita ambil kesimpulan bahwa mesin catur dengan *depth* 1 ini memiliki waktu komputasi yang rendah, dan tingkat kesulitan yang rendah sehingga pemain putih (manusia) dapat memenangkan permainan dengan mudah.

Depth Bermain 2

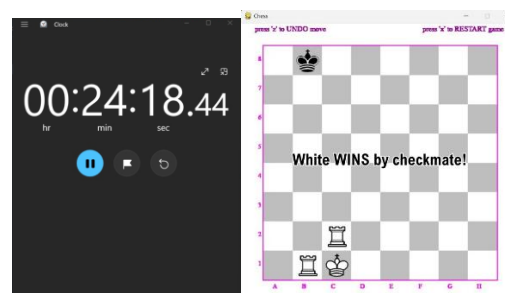


Gambar 13. *Depth* Bermain 2

Pada *depth* bermain 2, terdapat peningkatan dalam kedalaman pencarian pohon. Kondisi ini menunjukkan bahwa mesin kecerdasan buatan catur akan mencari lebih dalam lagi pada pohon pencarian. Tentu saja hal ini akan memakan waktu komputasi yang lebih banyak lagi

dikarenakan banyaknya jumlah akar yang harus dicari oleh mesin kecerdasan buatan. Hal tersebut akan menghasilkan mesin catur yang lebih sulit dikalahkan karena ia akan mencoba melihat beberapa probabilitas yang lebih menguntungkan. Pada Gambar 13. *Depth* Bermain 2, bisa dilihat bahwa perbedaan angka yang cukup signifikan apabila dibandingkan dengan hasil yang didapatkan pada *depth* bermain 1, ini merupakan suatu bukti bahwa dengan *depth* yang lebih tinggi maka akan menghasilkan mesin kecerdasan buatan catur yang lebih sulit lagi untuk dilawan oleh manusia.

- *Depth* Bermain 3



Gambar 14. *Depth* Bermain 3

Pada *depth* bermain 3, tingkat kesulitan dan jumlah waktu yang dibutuhkan untuk menyelesaikan permainan semakin bertambah, karena mesin catur akan mencari lebih banyak probabilitas yang bisa terjadi dalam setiap langkah yang akan diambil. Banyaknya akar yang perlu dicari oleh mesin kecerdasan buatan, membuat ia memakan waktu komputasi yang lebih banyak lagi sebelum akhirnya ia melakukan langkah. Tidak hanya waktu pencarian yang lebih lama, tingkat kesulitan mesin kecerdasan buatan juga akan meningkat sehingga menghasilkan lawan yang lebih sulit lagi untuk dikalahkan. Pada Gambar 14. *Depth* Bermain 3, waktu permainan memakan waktu yang lebih banyak lagi jika dibandingkan dengan *depth* 1 dan 2, hal ini terjadi karena waktu komputasi yang lama dan tingkat kesulitan yang tinggi membuat pemain putih (manusia) sedikit menunggu

dan kesulitan dalam menghadapi mesin kecerdasan buatan tersebut.

Pembahasan

Alpha-beta pruning merupakan sebuah algoritma yang memangkas cabang-cabang pohon pencarian yang tidak perlu, sehingga mempercepat pengambilan keputusan oleh mesin. Namun, kekurangan algoritma ini memerlukan lebih banyak tenaga komputasi untuk memproses pencarian pada kedalaman yang lebih besar. Sehingga apabila pengguna menggunakan komputer yang kurang support akan mengalami waktu pencarian yang lebih lama dari mesin kecerdasan buatan alpha-beta pruning.

Penggunaan alpha-beta pruning menawarkan fitur yang memungkinkan pengoptimalan pencarian untuk meningkatkan efisiensi komputasi. Alpha-beta pruning secara signifikan mengurangi jumlah node yang perlu dievaluasi, sehingga menghemat waktu dan sumber daya komputasi. Ini sangat krusial dalam permainan seperti catur, di mana pohon pencarian dapat menjadi sangat besar (Silver et al., 2017). Dari pernyataan tersebut, bisa kita tarik kesimpulan bahwa untuk menghemat sumber daya komputasi yang tinggi dari pohon pencarian catur yang besar, teknik algoritma alpha-beta pruning ini cukup efisien. Dengan memaksimalkan pencarian pada pohon, akan menghasilkan pengalaman bermain yang lebih baik bagi pengguna sekaligus juga membuktikan teori yang mengatakan bahwa dengan alpha-beta pruning dapat menjadi jawaban untuk memfasilitasi permainan strategi seperti catur ini.

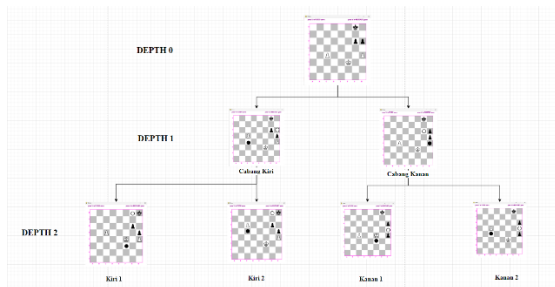
Berpacu kepada hasil yang penulis telah dapatkan, penerapan alpha-beta pruning ini memang memiliki kelebihan dan kekurangannya masing-masing apabila diterapkan pada permainan strategi seperti catur. Alpha-beta memanfaatkan nilai MiniMax untuk memangkas subpohon ketika memiliki bukti bahwa suatu langkah tidak akan mempengaruhi keputusan di simpul akar. Hal ini terjadi ketika pencarian

parsial pada subpohon mengungkapkan bahwa lawan memiliki kesempatan untuk menurunkan nilai MiniMax yang sudah ditetapkan yang didukung dari subpohon lain (Saffidine et al, 2012) Kelebihan alpha-beta pruning ini memungkinkan ia memangkas beberapa probabilitas dari pencarian akar pohon yang banyak, sedangkan kekurangannya adalah ia memerlukan waktu untuk memangkas semua probabilitas yang ada. Waktu komputasi yang dihasilkan dapat berbeda-beda tergantung dengan *software* dan *hardware* yang digunakan.

Pada permainan catur dengan *depth* 1 (satu), kedalaman berpikir mesin tidak terlalu dalam sehingga ketika pemain catur putih (manusia) melakukan langkah, mesin kecerdasan buatan langsung melakukan langkah selanjutnya secara singkat. Mesin tidak memakan waktu yang lama dikarenakan ia hanya mencari probabilitas dengan *depth* 1, hal ini mengakibatkan mesin catur menghasilkan output yang tidak maksimal. Berbanding terbalik apabila kita menggunakan kedalaman permainan catur menjadi *depth* 3 (tiga), hal tersebut akan memperlama durasi permainan catur dikarenakan mesin catur akan mencari probabilitas yang lebih dalam yang tentu saja akan memakan waktu yang lebih lama. Namun dengan *depth* yang lebih tinggi, tingkat kesulitan permainan catur akan menjadi lebih sulit, dan total waktu permainan akan menjadi lebih lama.

Dalam permainan catur yang telah penulis gunakan, bisa dilihat dengan level atau kedalaman yang semakin tinggi maka permainan akan memakan lebih banyak waktu lagi. Ini menjadi kekurangan utama dalam penggunaan alpha-beta pruning, algoritma ini mengharuskan komputer untuk menelusuri seluruh pohon, yang dapat membutuhkan banyak waktu dan kekuatan komputasi (Felstiner, 2019). Meskipun metode ini mengurangi jumlah simpul yang harus dievaluasi dibandingkan dengan pencarian *brute force* (algoritma untuk mengevaluasi seluruh kemungkinan gerakan untuk menentukan hasil yang

terbaik). Peningkatan kedalaman tetap mengakibatkan peningkatan jumlah simpul yang dievaluasi secara signifikan, sehingga memperpanjang waktu pencarian. Dalam implementasi praktis, hal ini mengharuskan penggunaan perangkat keras yang lebih kuat atau optimasi tambahan pada algoritma untuk menjaga waktu pemrosesan tetap wajar (Silver et al., 2016). Gambaran mengenai pemangkasan pohon alpha-beta pruning dapat dilihat pada Gambar 15. sebagai berikut:



Gambar 15. pohon alpha-beta pruning

Pada gambar pohon pencarian algoritma alpha-beta pruning ini merupakan probabilitas yang dapat terjadi dalam berjalannya permainan catur, keseluruhan probabilitas yang ada hanya akan diambil keputusan yang terbaik saja. Untuk mengetahui penjelasan langkah dapat dilihat pada penjabaran berikut:

Posisi Awal

Posisi catur awal bisa dilihat dari Gambar 16. Posisi Awal dimana pada gambar tersebut memiliki komposisi posisi catur : pion c4, pion h4, raja f3.



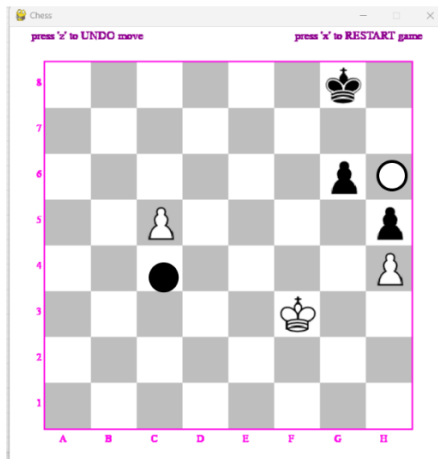
Gambar 16. Posisi Awal Langkah Putih (Manusia)

Pada kondisi ini, pemain manusia mendapatkan peran untuk memainkan catur putih. Pada gambar 16. Posisi Awal, pemain manusia memiliki beberapa langkah yang bisa diambil dari posisi tersebut. contoh pada gambar 16. pemain manusia (catur putih) memiliki beberapa kemungkinan untuk melangkahkan pion ataupun raja.

Langkah Hitam (Kecerdasan Buatan)

Mesin kecerdasan buatan catur memiliki peran untuk melangkah sebagai catur hitam. Mesin kecerdasan buatan (hitam) merespon dengan berbagai kemungkinan langkah dari posisi yang dihasilkan oleh manusia. Setelah manusia (catur putih) selesai melakukan langkah maka kecerdasan buatan akan mencari langkah terbaik dengan menggunakan algoritma alpha-beta pruning.

Cabang Kiri (Depth 1)



Gambar 17. Kiri (Depth 1)

Langkah Putih:

1. **Putih (c4 ke c5)** - Menggerakkan pion selangkah kedepan dimana pion berusaha untuk mencapai base hitam atau garis 8 agar bisa mendapatkan promosi.

Langkah Hitam:

2. **Hitam (h6 ke h5)** - setelah putih melangkahkan pion ke c5 hitam merespon dengan melangkahkan pionnya kedepan untuk memblokir langkah pion putih yang lain.

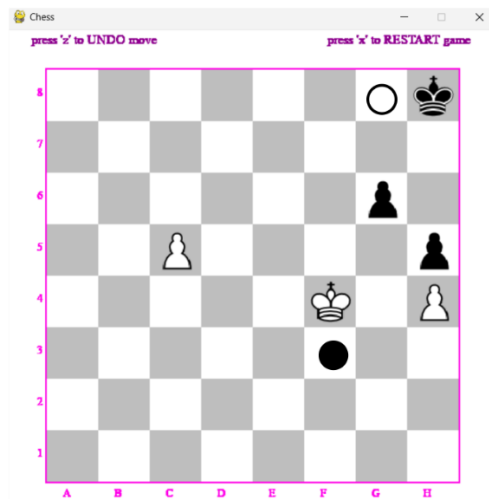
Langkah Putih:

1. **Putih (h4 ke h5)** - Menggerakkan pion selangkah kedepan dimana pion berusaha untuk mencapai base hitam atau garis 8 agar bisa mendapatkan promosi.

Langkah Hitam:

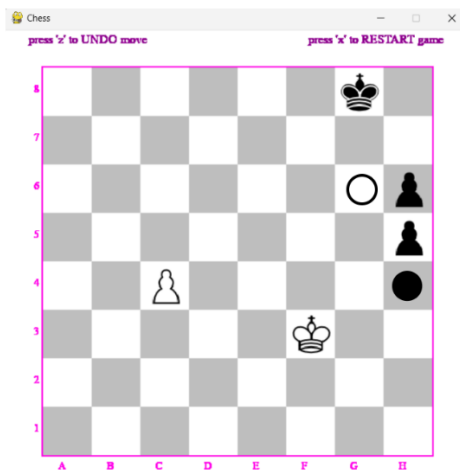
2. **Hitam (h6 ke h5)** - setelah putih melangkahkan pion ke h5. hitam merespon dengan melangkahkan pion hitam dari g6 ke h5 untuk memakan pion putih di h5.

Cabang Kiri 1 (Depth 2)



Gambar 19. Cabang Kiri 1 (Depth 2)

Cabang Kanan (Depth 1)



Gambar 18. Cabang Kanan (Depth 1)

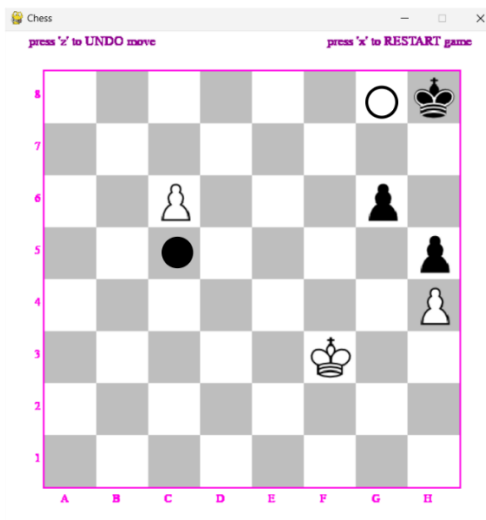
Langkah Putih:

1. **Putih (f3 ke f4)** - menggerakkan raja putih ke arah f5.

Langkah Hitam:

2. **Hitam (g8 ke h8)** - setelah putih melangkahkan raja ke f5. respon yang dilakukan oleh hitam adalah melakukan pergerakan rajanya dari g8 ke h8.

Cabang Kiri 2 (Depth 2)



Gambar 20. Cabang Kiri 2 (Depth 2)

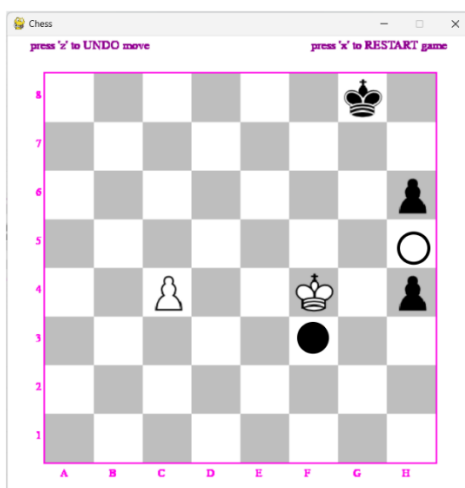
Langkah Putih:

1. **Putih (c5 ke c6)** - pion putih melakukan langkah ke f6 untuk mencapai promosi pada barisan 8.

Langkah Hitam:

2. **Hitam (g8 ke h8)** - setelah putih melangkahkan pion dari c5 ke c6 . respon yang dilakukan oleh hitam adalah melakukan pergerakan rajanya dari g8 ke h8.

Cabang Kanan 1 (Depth 2)



Gambar 21. Cabang Kanan 1 (Depth 2)

Langkah Putih:

1. **Putih (f3 ke f4)** - raja putih melakukan langkah ke f4.

Langkah Hitam:

2. **Hitam (h5 ke h4)** - setelah putih melangkahkan raja ke f4. respon yang dilakukan oleh catur hitam adalah melangkahkan pion dari f5 ke f4. Yang mana merupakan langkah terbaik yang dapat dilakukan oleh pihak catur hitam.

Cabang Kanan 2 (Depth 2)



Gambar 22. Cabang Kanan 2 (Depth 2)

Langkah Putih:

1. **Putih (c4 ke c5)** - pion putih melakukan langkah ke f5 agar semakin dekat dengan barisan 8 untuk mendapatkan promosi.

Langkah Hitam:

2. **Hitam (h5 ke h4)** - setelah putih melangkahkan pion ke c5. respon dari hitam adalah pion melakukan langkah dari h5 ke h4 yang mana itu merupakan langkah terbaik yang dapat dilakukan. Setiap langkah putih difokuskan pada pergerakan pion dengan tujuan untuk promosi atau membuka jalan bagi rajanya. Respons hitam (catur kecerdasan buatan) adalah menggerakkan pion untuk membuka jalur bagi raja atau untuk menghadapi ancaman dari

pion hitam. Pergerakan ini menyoroti strategi untuk mendekati promosi pion dan mempertahankan posisi raja.

Contoh Penerapan Alpha-Beta Pruning dalam Posisi yang Diberikan:

Langkah Putih (Manusia)

Pemain manusia (putih) memiliki beberapa langkah yang dapat diambil dari posisi awal tersebut. Contoh langkah yang mungkin adalah menggerakkan pion atau raja.

Langkah Hitam (Kecerdasan Buatan)

Kecerdasan buatan (hitam) merespons langkah putih dengan berbagai kemungkinan langkah dari posisi yang dihasilkan oleh putih. Setelah langkah putih selesai, kecerdasan buatan akan mencari langkah terbaik untuk hitam dan melakukannya.

Cabang Kiri (Depth 1)

Langkah Putih:

- Putih (c4 ke c5): Pion putih bergerak selangkah ke depan untuk mencapai baris 8 agar bisa mendapatkan promosi.

Langkah Hitam:

- Hitam (h6 ke h5): Pion hitam merespons dengan bergerak selangkah ke depan untuk memblokir langkah pion putih yang lain.

Penerapan Alpha-Beta:

- Di sini, langkah hitam yang memblokir langkah putih akan dievaluasi untuk melihat apakah ada langkah yang lebih baik. Jika langkah lain lebih baik, cabang ini bisa dipangkas.

Cabang Kanan (Depth 1)

Langkah Putih:

- Putih (h4 ke h5): Pion putih bergerak selangkah ke depan untuk mencapai baris 8 agar bisa mendapatkan promosi.

Langkah Hitam:

- Hitam (g6 ke h5): Pion hitam bergerak ke h5 untuk memakan pion putih di h5.

Penerapan Alpha-Beta:

- Evaluasi apakah langkah hitam memakan pion putih memberikan keuntungan maksimal. Jika langkah lain tidak memberikan keuntungan yang lebih besar, cabang tersebut bisa dipangkas.

Cabang Kiri 1 (Depth 2)

Langkah Putih:

- Putih (f3 ke f4): Raja putih bergerak ke f4.

Langkah Hitam:

- Hitam (g8 ke h8): Raja hitam bergerak ke h8.

Penerapan Alpha-Beta:

- Periksa apakah langkah raja hitam ini meminimalkan keuntungan putih. Jika langkah lain meminimalkan keuntungan lebih baik, cabang ini bisa dipangkas.

Cabang Kiri 2 (Depth 2)

Langkah Putih:

- Putih (c5 ke c6): Pion putih bergerak ke c6 untuk mendekati promosi di baris 8.

Langkah Hitam:

- Hitam (g8 ke h8): Raja hitam bergerak ke h8.

Penerapan Alpha-Beta:

- Evaluasi apakah langkah raja hitam ini adalah respons terbaik. Jika langkah lain memberikan hasil yang sama atau lebih baik, cabang ini bisa dipangkas.

Cabang Kanan 1 (Depth 2)

Langkah Putih:

- Putih (f3 ke f4): Raja putih bergerak ke f4.

Langkah Hitam:

- Hitam (h5 ke h4): Pion hitam bergerak dari h5 ke h4.

Penerapan Alpha-Beta:

- Tentukan apakah langkah hitam ini mencegah promosi pion putih dengan lebih baik. Jika tidak, cabang ini bisa dipangkas.

Cabang Kanan 2 (Depth 2)

Langkah Putih:

- Putih (c4 ke c5): Pion putih bergerak ke c5 untuk mendekati promosi di baris 8.

Langkah Hitam:

- Hitam (h5 ke h4): Pion hitam bergerak dari h5 ke h4.

Penerapan Alpha-Beta:

- Periksa apakah langkah hitam ini menghalangi promosi pion putih dengan optimal. Jika langkah lain lebih efektif, cabang ini bisa dipangkas.

Dalam setiap langkah, putih difokuskan pada pergerakan pion untuk mencapai promosi atau membuka jalan bagi rajanya. Respons hitam (kecerdasan buatan) adalah menggerakkan pion atau raja untuk menghadapi ancaman dari pion putih. Alpha beta pruning membantu memangkas cabang yang tidak perlu dievaluasi karena tidak mempengaruhi hasil akhir, sehingga mempercepat proses pencarian langkah terbaik. Algoritma ini dikategorikan cocok dengan permainan papan seperti catur, mengapa demikian? Karena dalam menganalisis hasil dari sebuah langkah permainan papan, hasil dari setiap probabilitas langkah kerap memiliki kualitas yang kurang atau bahkan sama dengan probabilitas langkah lainnya. Fungsi alpha-beta pruning disini adalah untuk memangkas langkah yang memiliki kualitas terburuk dan menghasilkan langkah terbaik dengan algoritma yang lebih efisien.

Sekarang muncul pertanyaan, apakah dengan *depth* yang tinggi akan menghasilkan kecerdasan buatan catur yang sangat sulit atau bahkan tidak mungkin dikalahkan oleh manusia? Jawabannya adalah, belum tentu. Pada video yang diunggah oleh saluran YouTube *WIRED* “*Why AI Chess Bots Are Virtually Unbeatable (ft. GothamChess) | WIRED*” disebutkan dengan kecanggihan teknologi sekarang, mesin kecerdasan buatan catur dapat melihat puluhan ribu probabilitas tepat saat permainan catur dimulai. Meskipun demikian, manusia tetap akan memiliki kesempatan untuk menang melawan mesin kecerdasan buatan, dikarenakan mesin kecerdasan buatan tetap memiliki kemungkinan untuk melakukan

kesalahan. Manusia tetap saja akan mengalami kesulitan untuk menghadapi mesin kecerdasan buatan catur tersebut.

UCAPAN TERIMA KASIH

Kami selaku penulis mengucapkan terima kasih kepada semua pihak yang telah dukungan dan bantuan selama proses penyusunan jurnal ini. Secara Khusus, kami ingin menyampaikan rasa terima kasih kepada:

1. Drs. Anggraini Puspita Sari, yang telah memberikan bimbingan, saran, dan dorongan yang sangat berarti dalam penyusunan jurnal ini. Arahan dan pengetahuan yang sangat membantu memperdalam pemahaman tentang topik yang dibahas.
2. Universitas UPN “Veteran” Jawa Timur, khususnya Fakultas Ilmu Komputer, yang telah menyediakan fasilitas sumber daya yang diperlukan untuk menyelesaikan penelitian ini.
3. Teman-teman Kelompok 10, yang telah bekerja sama dengan baik dalam berdiskusi dan bertukar ide yang aspiratif. Kontribusi masing-masing anggota kelompok sangat penting dalam penyelesaian jurnal ini.
4. Keluarga dan sahabat, yang selalu memberikan dukungan moral dan semangat selama proses penelitian dan penulisan jurnal ini.
5. Warung kopi Pusat dan Kopi Janji Jiwa yang telah memfasilitasi kami untuk menyelesaikan Jurnal.
6. Semua pihak yang tidak dapat saya sebutkan satu per satu, namun telah memberikan dukungan dalam bentuk apapun, baik secara langsung maupun tidak langsung.

Terima kasih kepada semuanya yang telah membantu penyusunan jurnal ini. Semoga hasil dari penelitian ini dapat memberikan kontribusi yang bermanfaat bagi ilmu pengetahuan dan pengembangan lebih lanjut di bidang teknologi.

KESIMPULAN

Berdasarkan hasil evaluasi terhadap penerapan algoritma Alpha-Beta pruning

dalam permainan catur, penelitian ini menyimpulkan bahwa algoritma ini secara signifikan meningkatkan efisiensi pencarian langkah optimal. Alpha-beta pruning mampu memangkas cabang-cabang yang tidak relevan dalam pohon pencarian, sehingga mengurangi jumlah simpul yang perlu dievaluasi dan menghemat waktu serta sumber daya komputasi. Pada *depth* pencarian yang rendah, waktu komputasi relatif singkat dan tingkat kesulitan permainan lebih rendah. Namun, ketika *depth* pencarian meningkat, waktu komputasi menjadi lebih lama dan tingkat kesulitan permainan meningkat secara signifikan. Sebagai contoh, pada kedalaman pencarian 1, mesin catur menghasilkan hasil yang cepat namun dengan tingkat kesulitan rendah, sementara pada *depth* pencarian 3, terjadi peningkatan signifikan dalam waktu komputasi serta tingkat kesulitan permainan. Dapat kita tarik kesimpulan dari pernyataan tersebut bahwa dengan *depth* yang semakin tinggi maka akan menghasilkan mesin kecerdasan buatan yang lebih teliti lagi untuk mencari probabilitas-probabilitas yang tidak akan dicapai dengan kemampuan berpikir manusia. Namun bukan berarti suatu mesin kecerdasan buatan tidak dapat dikalahkan, mesin kecerdasan buatan tetap bisa melakukan kesalahan dalam proses komputasi.

Penggunaan alpha-beta pruning dalam pengembangan kecerdasan buatan untuk permainan catur sangat efisien dalam mengurangi jumlah simpul yang perlu dievaluasi, yang secara langsung menghemat waktu dan sumber daya komputasi. Meskipun demikian, peningkatan kedalaman pencarian memerlukan perangkat keras yang lebih kuat atau optimasi tambahan pada algoritma untuk menjaga waktu pemrosesan tetap wajar. Secara keseluruhan, penelitian ini membuktikan bahwa alpha-beta pruning adalah teknik yang efektif untuk meningkatkan efisiensi dalam pencarian langkah optimal dalam permainan catur. Hasil penelitian ini

diharapkan dapat menjadi acuan dalam pengembangan lebih lanjut dalam bidang kecerdasan buatan permainan catur dan aplikasi strategi permainan lainnya.

DAFTAR PUSTAKA

- Brooks, J. (2022). The Art of Problem Solving And Its Translation Into Practice. Diambil dari <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9442556/>
- Fédération Internationale des Échecs (FIDE). (2020). Laws of Chess. Diambil dari <https://www.fide.com/FIDE/handbook/LawsOfChess.pdf>
- Felstiner, C. (2019, May 9). Alpha-Beta Pruning. Diambil dari <https://www.whitman.edu/documents/Academics/Mathematics/2019/Felstiner-Guichard.pdf>
- Gobet, F., & Campitelli, G. (2006). Educational Benefits of Chess Instruction: A critical review. Diambil dari https://researchportal.murdoch.edu.au/esploro/fulltext/bookChapter/Educational-benefits-of-chess-instruction-A/991005542720507891?repId=12135852430007891&mid=1313670366007891&institution=61MUN_INST
- Indra, E., Christnatalis, Sijabat, N.P., Riady, M.A., & Lumbantombing, J.S.M. (2020). Analisa Efektivitas Algoritma Minimax, Alpha Beta Pruning, dan Negamax dalam Penerapannya pada Permainan Papan (Board Game). Diambil dari <http://ejournal.sisfokomtek.org/index.php/jikom/article/view/48/52>
- Nanu, C.C., Coman, C., Bularca, M.C., Schmitz, L.M., Gotea, M., Atudorei, I., Turcu, I., & Negrila, I. (2023). The Role of Chess In The Development of Children-Parents' Perspectives. Diambil dari <https://www.frontiersin.org/articles/10.3389/fpsyg.2023.1210917/full>
- Ribeiro, A.C., Rios, L.M., Gomes, R.M., Faria, B.M., & Reis, L.P. (2017). Data

- mining in adversarial search — Players Movement Prediction In Connect 4 Games. Diambil dari https://www.researchgate.net/profile/Ana-Ribeiro-56/publication/318417488_Data_mining_in_adversarial_search_-_players_movement_prediction_in_connect_4_games/links/5c1bfcf2a6fdccfc705da5fe/Data-mining-in-adversarial-search-players-movement-prediction-in-connect-4-games.pdf
- Root, A. W. (2008). *Children and Chess: A Guide for Educators*. Diambil dari https://books.google.co.id/books?hl=id&lr=&id=CbvOEAAAQBAJ&oi=fnd&pg=PR7&dq=Root,+A.+W.,+%22Children+and+Chess:+A+Guide+for+Educators,+2008.&ots=auIgAgfTf4&sig=TT1hpRGshIqa5KCeYjRCGjNHMU&redir_esc=y#v=onepage&q&f=false
- Saffidine, A., Finnsen, H., & Buro, M. (2012). Alpha-Beta Pruning for Games with Simultaneous Moves. Diambil dari <https://ojs.aaai.org/index.php/AAAI/article/download/8148/8006>
- Sari, A.P., Prasetya, D.A., Haromainy, M.M.A., Aditiawan, F.P., Sihananto, A.N., & Saputra, W.S.J. (2022). Analisis Faktor Kesuksesan Penggunaan eBelajar Menggunakan Metode Hot-Fit di STIKI Malang. Diambil dari <https://prosiding-senada.upnjatim.ac.id/index.php/senada/article/download/54/16>
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489. <https://doi.org/10.1038/nature16961>
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv preprint arXiv:1712.01815*. Diambil dari <https://arxiv.org/abs/1712.01815>
- Singhal, S.P. & Sridevi, M. (2019). Comparative Study Of Performance Of Parallel Alpha Beta Pruning For Different Architectures. Diambil dari <https://arxiv.org/pdf/1908.11660>
- Utami, W. Y. D. (2013). Meningkatkan Minat Belajar Matematika Melalui Permainan Teka-Teki. Diambil dari <https://journal.unj.ac.id/unj/index.php/jiv/article/download/7694/5504>
- Wicaksono, A.I., Ardianto, D.T., & Erandaru, E. (2019). Perancangan Boardgame Augmented Reality Sebagai Media Pembelajaran Bertema Perang Kerajaan Kediri Melawan Tentara Mongol. diambil dari <https://publication.petra.ac.id/index.php/dkv/article/viewFile/8667/7824>