Analisis Algoritma Langkah Optimal dalam Catur dengan Pendekatan Pohon dan Kombinatorika

Nathanael Rachmat - 13523142[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*nathanael.rachmat@gmail.com*, *13523142@std.stei.itb.ac.id*

***Abstraksi*— Dalam permainan catur, proses pengambilan langkah optimal merupakan aspek penting yang membutuhkan analisis mendalam terhadap posisi papan dan kemungkinan gerakan lawan. Penelitian ini bertujuan untuk menganalisis algoritma langkah optimal dalam permainan catur menggunakan pendekatan pohon keputusan dan kombinatorika. Dalam pendekatan ini, posisi catur direpresentasikan sebagai simpul dalam pohon, di mana setiap cabang merepresentasikan gerakan potensial. Evaluasi dilakukan dengan menghitung nilai evaluasi (evaluation bar) pada setiap simpul, yang kemudian digunakan untuk menentukan jalur terbaik dari akar ke daun berdasarkan total evaluasi. Untuk efisiensi, algoritma mengimplementasikan fungsi kombinasi untuk menghasilkan semua gerakan legal pada setiap posisi, serta pendekatan rekursif untuk membangun pohon hingga kedalaman tertentu. Hasil dari penelitian ini menunjukkan bahwa algoritma yang diusulkan mampu menemukan jalur optimal secara efektif berdasarkan nilai evaluasi posisi. Pendekatan ini dapat diimplementasikan dalam pengembangan sistem catur berbasis komputer dan pembelajaran strategi permainan.**

***Kata kunci*—Catur, Algoritma Langkah Optimal, Pohon Keputusan, Kombinatorka, Evaluasi Posisi**

I. PENDAHULUAN

Catur merupakan salah satu permainan strategi yang telah populer di berbagai belahan dunia selama berabad-abad. Tidak hanya menjadi hiburan semata, catur juga digunakan sebagai alat pembelajaran untuk melatih kemampuan berpikir logis, analitis, dan pengambilan keputusan. Dalam kehidupan sehari-hari, catur sering kali menjadi simbol kompetisi intelektual yang melibatkan berbagai tingkat strategi, dari amatir hingga profesional, termasuk dalam kejuaraan dunia yang sangat prestisius.

Permainan catur memiliki sifat unik dengan kompleksitas yang luar biasa. Meskipun aturan dasar catur relatif sederhana, jumlah kombinasi langkah yang mungkin dari setiap posisi sangatlah besar. Diperkirakan terdapat sekitar 10^{120} posisi legal yang dapat tercipta sepanjang permainan, angka yang dikenal sebagai Shannon Number. Angka ini bahkan melampaui jumlah atom di alam semesta yang diketahui, menunjukkan betapa rumitnya permainan ini dari perspektif matematika dan komputer.

Dalam dunia informatika, permainan catur sering kali direpresentasikan menggunakan struktur pohon. Setiap posisi pada papan catur dianggap sebagai simpul (node), sedangkan setiap langkah legal yang mungkin diambil dari posisi tersebut menjadi cabang (branch). Dengan pendekatan pohon ini, kita dapat memvisualisasikan seluruh kemungkinan jalur permainan mulai dari akar (root) hingga simpul daun (leaf), yang masing-masing merepresentasikan hasil akhir dari suatu urutan langkah.

Selain pohon, permainan catur juga erat kaitannya dengan kombinatorika, cabang matematika yang mempelajari penghitungan, penyusunan, dan pemilihan objek dalam suatu himpunan. Dalam konteks catur, kombinatorika digunakan untuk menentukan semua kemungkinan langkah legal pada setiap posisi, menganalisis pola permainan, serta memprediksi jalur optimal berdasarkan evaluasi posisi. Pendekatan kombinatorika ini memungkinkan analisis yang lebih sistematis terhadap berbagai strategi, yang menjadi landasan dalam pengembangan algoritma untuk permainan catur berbasis komputer.

Penelitian ini bertujuan untuk menganalisis algoritma langkah optimal dalam permainan catur dengan memanfaatkan pendekatan pohon dan kombinatorika. Melalui implementasi algoritma ini, diharapkan dapat memberikan pemahaman yang lebih mendalam terhadap pengambilan keputusan optimal dalam permainan catur serta aplikasinya dalam pengembangan teknologi berbasis kecerdasan buatan.

II. DASAR TEORI

Dalam permainan catur, terdapat berbagai macam dasar teori yang bisa diterapkan. Untuk mengetahui bagaimana algoritma langkah optimal dalam catur dengan pendekatan pohon dan kombinatorika, diperlukan beberapa dasar teorinya. Berikut ini merupakan beberapa dasar teori yang penting dalam penelitian ini:

A. Permainan Catur dan Kompleksitasnya

Catur adalah permainan strategi dua pemain yang dimainkan di papan berukuran 8x8 kotak dengan total 64 petak berwarna terang dan gelap secara bergantian. Setiap pemain memulai dengan 16 buah catur yang terdiri dari: 1 Raja, 1 Ratu, 2 Benteng, 2 Kuda, 2 Gajah, 8 Pion.

Tujuan utama permainan ini adalah melakukan skakmat terhadap raja lawan, yaitu posisi di mana raja berada dalam ancaman serangan dan tidak ada langkah legal yang dapat menghindarinya. Setiap jenis buah catur memiliki pola gerakan unik: raja dapat bergerak satu petak ke segala arah, ratu dapat

bergerak ke segala arah sejauh mungkin (diagonal, vertical dan horizontal), benteng dapat bergerak horizontal atau vertikal sejauh mungkin, gajah dapat bergerak diagonal sejauh mungkin, kuda dapat bergerak dalam pola 'L', yaitu dua petak dalam satu arah dan satu petak tegak lurus, serta pion dapat bergerak satu petak ke depan; pada langkah pertama dapat bergerak dua petak; menangkap lawan secara diagonal.

Permainan catur memiliki kompleksitas yang sangat tinggi. Jumlah kemungkinan posisi legal dalam catur diperkirakan mencapai 10^120, yang dikenal sebagai *Angka Shannon*. Hal ini menunjukkan bahwa meskipun aturan dasarnya sederhana, jumlah kombinasi langkah yang mungkin sangatlah besar, membuat analisis dan prediksi dalam permainan catur menjadi tantangan yang signifikan.

B. Algoritma Evaluasi dalam Mesin Catur

Mesin catur menggunakan algoritma evaluasi untuk menilai posisi di papan dan menentukan langkah terbaik. Evaluasi ini biasanya dinyatakan dalam bentuk nilai numerik, di mana nilai positif menunjukkan keunggulan bagi putih dan nilai negatif menunjukkan keunggulan bagi hitam. Beberapa faktor yang dipertimbangkan dalam evaluasi meliputi:

- Materi: nilai total buah catur di papan.
- Mobilitas: jumlah langkah legal yang tersedia.
- Keamanan Raja: seberapa terlindunginya raja dari serangan.
- Struktur Pion: formasi pion dan kelemahannya.
- Kontrol Spasi: siapa yang lebih menguasai papan caturnya.
- Koordinasi bidak: seberapa baik bidak catur bekerja sama.

Algoritma evaluasi dapat bervariasi dalam kompleksitasnya. Misalnya, mesin catur Stockfish menggunakan pendekatan evaluasi yang kompleks dengan mempertimbangkan berbagai faktor posisi secara mendalam. Panduan evaluasi Stockfish memberikan wawasan mendalam tentang bagaimana mesin ini menilai berbagai aspek posisi catur.

Dalam konteks makalah ini, algoritma evaluasi ini merupakan inti dari pohon permainan. Setiap simpul dihitung menggunakan fungsi evaluasi, dan nilai ini digunakan untuk membandingkan langkah-langkah dalam kombinatorika. Hal ini memungkinkan pemain (atau mesin) untuk memilih langkah optimal berdasarkan kombinasi aspek strategis dan material.

C. Pohon dan Hubungannya dengan Catur

Pohon adalah graf tak-berarah yang terhubung dan tidak mengandung siklus. Syarat utama dari pohon mencakup:

- Tidak berarah: Semua sisi tidak memiliki arah tertentu.
- Terhubung: Semua simpul (node) dalam graf saling terhubung.
- Tanpa siklus: Tidak ada lintasan tertutup dalam graf.

Properti penting dari pohon meliputi:

- Pohon memiliki n simpul dan n – 1 sisi.
- Setiap pasangan simpul dalam pohon hanya terhubung dengan satu lintasan unik.
- Penambahan satu sisi baru pada pohon akan menghasilkan satu siklus

Pohon berakar adalah pohon di mana salah satu simpul dipilih sebagai akar. Pada pohon ini, semua sisi dianggap memiliki arah yang menuju atau menjauhi akar. Pohon berakar sering digunakan dalam algoritma rekursif untuk merepresentasikan hubungan hierarkis

Terminologi pohon berakar meliputi:

- Anak (child): Simpul yang berada satu tingkat lebih dalam dari simpul induknya.
- Orangtua (parent): Simpul yang berada satu tingkat lebih dekat ke akar.
- Daun (leaf): Simpul yang tidak memiliki anak.
- Tingkat (level): Jarak sebuah simpul dari akar.

Dalam permainan catur, setiap posisi dapat direpresentasikan sebagai simpul pada pohon. Transisi dari satu posisi ke posisi berikutnya (berdasarkan langkah catur yang mungkin) direpresentasikan sebagai cabang pada pohon. Kedalaman pohon merepresentasikan jumlah langkah ke depan yang dapat dianalisis, di mana setiap simpul menyimpan evaluasi dari posisi tersebut.

Pohon digunakan dalam algoritma evaluasi posisi catur seperti Minimax, alpha-beta pruning, dan evaluasi probabilistik. Dengan memanfaatkan sifat rekursif pohon berakar, algoritma dapat menentukan langkah optimal hingga kedalaman tertentu (depth). Kombinasi antara pohon dan teori kombinatorika memberikan pendekatan sistematis untuk menganalisis strategi langkah dalam permainan catur

D. Kombinatorika dan Hubungannya dengan Catur

Kombinatorika adalah cabang matematika yang mempelajari cara menghitung jumlah pengaturan atau pemilihan objek-objek tanpa harus menyusun seluruh kemungkinan satu per satu. Kombinatorika mencakup prinsip dasar seperti rule of sum dan rule of product, serta konsep permutasi dan kombinasi. Sebagai contoh, permutasi menghitung jumlah pengaturan elemen di mana urutan elemen diperhitungkan, sedangkan kombinasi menghitung jumlah pengaturan di mana urutan elemen tidak diperhitungkan

Dalam permainan catur, setiap langkah pemain menghasilkan posisi baru dengan berbagai kemungkinan gerakan. Kombinasi ini dapat dihitung menggunakan prinsip kombinatorika. Permainan catur memiliki sekitar 10^{120}kemungkinan posisi unik, yang dikenal sebagai *number of chess positions*. Angka ini jauh melampaui jumlah atom di alam semesta, menggambarkan kompleksitas kombinatorik catur.

Setiap gerakan dalam catur dapat dianggap sebagai permutasi dari langkah-langkah sebelumnya, di mana aturan permainan dan posisi papan membatasi ruang kemungkinan. Contohnya, jika setiap bidak memiliki n kemungkinan langkah dan ada p bidak aktif, maka langkah keseluruhan dapat dihitung dengan prinsip perkalian $p \times n$.

Dalam catur, banyak situasi yang melibatkan pengulangan pola atau posisi tertentu. Prinsip kombinasi dengan pengulangan dapat digunakan untuk menghitung situasi ini. Sebagai contoh, jika seorang pemain ingin menghitung jumlah kemungkinan langkah yang melibatkan langkah bidak (pawn), gajah (bishop), atau kuda (knight), prinsip inklusi-eksklusi membantu

menghitung langkah unik dengan menghindari penghitungan langkah yang berulang.

Dalam analisis langkah optimal, kombinatorika digunakan untuk menjelajahi berbagai jalur permainan dalam pohon keputusan (decision tree). Setiap simpul pohon mewakili posisi papan tertentu, dan cabang-cabangnya mewakili langkah-langkah potensial. Dengan menerapkan kombinatorika, ruang langkah-langkah ini dapat diorganisasikan dan dievaluasi secara efisien untuk menemukan langkah optimal.

E. Kedalaman dalam Pohon Catur

Dalam analisis mesin catur, "kedalaman" merujuk pada seberapa jauh mesin menganalisis langkah-langkah ke depan, diukur dalam setengah langkah atau ply. Semakin dalam analisis, semakin akurat prediksi langkah optimalnya. Namun, peningkatan kedalaman memerlukan sumber daya komputasi yang lebih besar. Mesin seperti Stockfish dan Leela Chess Zero (LCZero) memiliki pendekatan berbeda dalam hal kedalaman analisis. Stockfish cenderung mencapai kedalaman lebih tinggi dengan menganalisis banyak kemungkinan langkah per detik, sementara LCZero fokus pada evaluasi posisi yang lebih mendalam meskipun dengan kedalaman yang lebih rendah.

Memahami konsep kedalaman ini penting untuk menilai kualitas analisis yang diberikan oleh mesin catur dan bagaimana mereka mencapai kesimpulan tentang langkah terbaik dalam suatu posisi. Oleh karena itu, karena daya komputasi yang terbatas, kedalaman dari implementasi pohon catur tidak akan melebihi dua agar hemat dalam waktu penelitian.

III. METODOLOGI

A. Permodelan Pohon

Pohon permainan (game tree) adalah representasi grafis dari semua kemungkinan langkah dalam permainan catur.

Pohon permainan dimulai dari posisi awal papan catur, yang menjadi simpul akar (root node). Posisi awal ini memuat informasi tentang letak bidak, giliran pemain (putih atau hitam), dan status permainan (misalnya, hak rokade atau en passant).

Setiap langkah legal dari posisi awal menjadi cabang dari akar pohon. Cabang ini meluas hingga simpul berikutnya yang merepresentasikan posisi baru setelah langkah tersebut diambil. Proses ini berlanjut untuk setiap posisi hingga mencapai kedalaman tertentu (depth), yaitu jumlah langkah ke depan yang ingin dievaluasi.

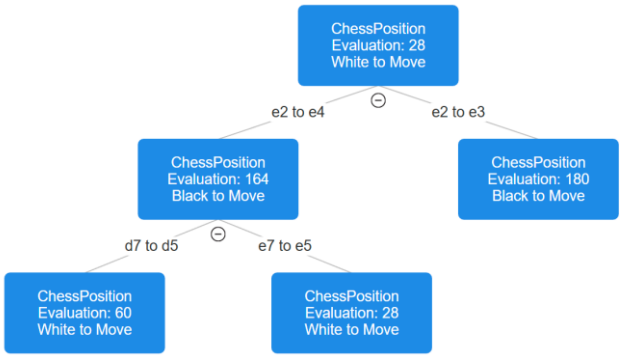
Struktur Pohon yang digunakan dalam penelitian ini dijelaskan sebagai berikut. Pohon utama Pohon didefinisikan dengan tipe data ChessTree, yang merupakan alamat simpul akar (NodeAddress). Simpul pohon (ChessTreeNode) memodelkan satu posisi catur (ChessPosition). Simpul memiliki daftar anak (children) yang merepresentasikan semua langkah legal dari posisi tersebut. Simpul Anak (ChildNode) adalah bagian dari daftar terhubung untuk menyimpan referensi ke simpul anak. *Info* menunjuk ke simpul anak, dan next menunjuk ke simpul anak berikutnya.

Model dari simpul catur direpresentasikan sebagai ChessPosition. Struktur ini merepresentasikan keadaan papan dan semua atribut yang relevan pada suatu posisi dalam permainan sebagai berikut: Papan Catur (b[8][8]) Array 2D (8x8) yang menyimpan posisi semua bidak, Hak Castling (c[4]) yang menyimpan informasi tentang apakah castling masih diperbolehkan untuk kedua sisi, Target En Passant (e[2]) yang menyimpan koordinat target en passant (jika ada). Giliran Pemain (w), yaitu boolean yang menunjukkan giliran pemain (true untuk putih, false untuk hitam), Counter Langkah (m[2]) menyimpan jumlah langkah permainan untuk keperluan aturan 50 langkah, Evaluasi Posisi (evaluationBar) yang menyimpan nilai evaluasi posisi saat ini, digunakan untuk menentukan langkah optimal, dan Langkah Terakhir (lastMove) yang menyimpan langkah terakhir yang dilakukan.

Pohon permainan catur dibangun dengan akar pohon (root) mewakili posisi awal permainan catur (ChessPosition). Setiap langkah legal dari posisi akar menghasilkan simpul anak. Simpul anak disimpan dalam daftar terhubung menggunakan ChildNode. Setiap simpul anak akan memodelkan langkah-langkah legal dari posisinya, membentuk cabang baru hingga kedalaman tertentu. Setiap simpul dihitung nilai evaluasinya (evaluationBar). Pohon ini digunakan untuk menentukan langkah optimal dengan menjelajahi simpul-simpul dan memilih cabang dengan nilai evaluasi terbaik.

Depth dari pohon mengontrol seberapa jauh langkah-langkah masa depan dieksplorasi. Semakin dalam eksplorasi, semakin banyak kemungkinan posisi yang perlu dianalisis, sehingga membutuhkan waktu komputasi yang lebih besar.

Pohon ini dibangun secara rekursif. Pada setiap simpul, semua langkah legal dihitung menggunakan kombinatorika, dan setiap langkah menghasilkan simpul baru. Simpul ini kemudian dihubungkan sebagai anak dari simpul induknya.



Gambar 1. Contoh permodelan pohon permainan catur.

B. Penerapan Kombinatorika

Kombinatorika digunakan untuk menghitung semua kemungkinan langkah legal dalam setiap posisi. Kombinatorika membantu menghasilkan daftar langkah legal dari setiap bidak, seperti pion, kuda, atau ratu. Aturan permainan digunakan sebagai dasar untuk menentukan validitas langkah. Dalam posisi tertentu, kombinatorika menghitung permutasi langkah dari semua bidak yang memungkinkan. Misalnya, jika terdapat 5 bidak dengan masing-masing 5 langkah legal, maka ada $5^5 = 3125$ kombinasi yang harus dievaluasi. Setelah semua langkah dihasilkan, langkah-langkah ini diberi peringkat berdasarkan evaluasi posisi yang dihasilkan. Langkah dengan evaluasi tertinggi untuk pemain putih (atau terendah untuk pemain hitam) akan diprioritaskan untuk eksplorasi lebih lanjut.

Setiap fungsi (pawn_moves, rook_moves, dll.) mengidentifikasi langkah legal untuk bidak tertentu dengan mempertimbangkan aturan langkah dan kondisi papan saat ini. Dalam validasi awal, pastikan bidak yang dimaksud adalah milik pemain yang sedang bergerak. Validasi dilakukan dengan mengecek apakah karakter bidak pada posisi awal sesuai giliran (putih atau hitam). Setelah itu dilakukannya iterasi langkah legal. Untuk menteri (queen_moves), gajah (bishop_moves), dan benteng (rook_moves), iterasi dilakukan berdasarkan arah langkah (horizontal, vertikal, diagonal). Untuk kuda (knight_moves), langkah legal ditentukan oleh offset koordinat tetap (8 kemungkinan). Untuk pion (pawn_moves), langkah legal mencakup langkah maju satu kotak, dua kotak dari baris awal, dan tangkapan diagonal (termasuk en passant). Setelah itu setiap langkah diperiksa apakah berada dalam batas papan (0–7) dan apakah kotak tujuan tidak ditempati oleh bidak sendiri. Jika kotak tujuan berisi bidak lawan, langkah ditambahkan, tetapi iterasi dihentikan untuk langkah menteri, benteng, atau gajah. Langkah yang valid disimpan dalam array Move dengan alokasi memori dinamis, yang dapat diubah ukurannya (realloc) sesuai kebutuhan.

Fungsi *combination* menggabungkan semua langkah legal yang dihasilkan oleh fungsi-fungsi individual untuk masing-masing bidak. Setiap kotak di papan diperiksa untuk mendeteksi bidak yang dimiliki pemain yang sedang bergerak. Berdasarkan jenis bidak ('P', 'R', 'N', 'B', 'Q', 'K'), langkahnya dihitung menggunakan fungsi terkait. Langkah-langkah yang dihasilkan digabungkan ke dalam array *combinedList* dengan alokasi ulang memori dinamis. Fungsi ini menghasilkan array langkah lengkap untuk posisi saat ini, yang dapat digunakan untuk eksplorasi pohon atau analisis langkah optimal. Kombinasi langkah-langkah legal ini merupakan dasar untuk membangun pohon permainan. Setiap simpul pohon memuat posisi baru yang dihasilkan dari satu langkah legal, dan simpul-simpul anak mencerminkan langkah-langkah legal dari posisi tersebut. Hal ini memungkinkan eksplorasi eksponensial semua kemungkinan jalur permainan, hingga kedalaman tertentu.

C. Algoritma Evaluasi

Algoritma evaluasi bertujuan untuk memberikan nilai numerik pada posisi catur berdasarkan berbagai aspek strategis dan material. Nilai evaluasi ini digunakan untuk menentukan langkah terbaik dalam permainan catur. Berikut adalah penjelasan rinci berdasarkan kode yang diberikan.

Evaluasi utama menghitung nilai kombinasi dari berbagai aspek permainan, termasuk evaluasi middle-game, end-game, dan faktor lainnya seperti aturan 50 langkah. Evaluasi *Middle-Game* mengukur posisi berdasarkan faktor-faktor yang dominan di tengah permainan, seperti mobilitas, keamanan raja, dan struktur pion. Evaluasi *End-Game* Mengukur posisi berdasarkan faktor yang penting di akhir permainan, seperti pion yang lewat dan jarak raja ke pion. Fase permainan diukur untuk menentukan seberapa besar kontribusi middle-game dibandingkan end-game. Fase dihitung berdasarkan jumlah material di papan. Evaluasi end-game dikalibrasi menggunakan faktor skala yang memperhitungkan keadaan spesifik dari posisi. Nilai akhir dihitung dengan menggabungkan evaluasi middle-game dan end-game menggunakan nilai fase sebagai pembobot. Nilai akhir disesuaikan berdasarkan aturan 50 langkah (50-move rule). Semakin mendekati aturan ini, semakin kecil nilai evaluasi untuk mendorong langkah yang mempercepat kemenangan. Tambahan nilai untuk pemain yang memiliki giliran bermain (tempo).

Dalam Evaluasi *Middle-Game* ataupun *Endgame*, terdapat beberapa ukuran yang dihitung. Nilai material dihitung untuk setiap bidak. Misalnya, ratu = 9, benteng = 5, dan pion = 1. Selisih nilai material antara putih dan hitam menjadi salah satu faktor utama evaluasi. Nilai Tabel Piece-Square dihitung berdasarkan posisi setiap bidak pada papan. Misalnya, pion lebih bernilai di pusat papan dibandingkan di sisi. Nilai ketidakseimbangan material adalah nilai dalam jenis bidak (misalnya, memiliki dua kuda lebih baik daripada satu benteng). Nilai struktur pion juga dihitung seperti pion yang terisolasi, pion yang terbelakang, dan pion yang terhubung dihitung. Nilai mobilitas dihitung dengan jumlah langkah legal yang tersedia untuk setiap bidak. Nilai ancaman merepresentasikan ancaman terhadap awan. Nilai pion lewat merepresentasikan pion yang memiliki jalan bebas menuju promosi dihitung sebagai faktor positif. Nilai keamanan raja dievaluasi berdasarkan tingkat perlindungan dari posisi raja. Nilai ruang merepresentasikan area yang dikuasai oleh pemain di papan. Nilai kemungkinan kemenangan merepresentasikan posisi yang memiliki kemungkinan besar untuk menang mendapatkan poin tambahan.

Evaluasi ini digunakan pada setiap simpul dalam pohon permainan untuk menentukan nilai posisi. Simpul dengan evaluasi tertinggi diprioritaskan untuk eksplorasi lebih lanjut.

Contoh perhitungan:

- Jika posisi saat ini memiliki keunggulan material besar (misalnya, memiliki ratu ekstra), evaluasi middle-game memberikan nilai tinggi.
- Jika posisi mendekati end-game dengan pion lepas, evaluasi end-game memberikan nilai tinggi pada pion tersebut.

D. Implementasi Algoritma

Implementasi algoritma dalam penelitian ini dilakukan dengan menggunakan struktur data dan algoritma yang dirancang secara efisien untuk merepresentasikan posisi, langkah, dan pohon permainan catur. Struktur data utama yang digunakan meliputi: ChessPosition, yang menyimpan informasi lengkap tentang posisi permainan, termasuk giliran pemain, hak rokade, en passant, serta evaluasi posisi; Move, yang merepresentasikan langkah dengan menyimpan koordinat posisi awal dan tujuan; dan Tree, yang digunakan untuk memodelkan permainan sebagai pohon, di mana setiap simpul menyimpan posisi catur dan setiap cabang merepresentasikan langkah legal yang dapat diambil dari posisi tersebut. Struktur pohon ini memungkinkan eksplorasi langkah-langkah permainan secara sistematis dan rekursif.

Eksplorasi pohon permainan dilakukan dengan algoritma rekursif menggunakan fungsi-fungsi yang menghasilkan kombinasi langkah legal untuk setiap jenis bidak, seperti queen_moves, knight_moves, bishop_moves, dan lainnya. Setiap simpul pohon menyimpan informasi langkah terakhir dan evaluasi posisi berdasarkan algoritma evaluasi yang menggabungkan parameter seperti nilai material, mobilitas, struktur bidak, dan keamanan raja. Kedalaman eksplorasi pohon dapat disesuaikan berdasarkan kebutuhan, misalnya hingga

kedalaman 2 untuk permainan cepat atau hingga kedalaman 20 untuk analisis mendalam. Pembatasan dilakukan berdasarkan waktu atau ruang komputasi yang tersedia, sehingga algoritma tetap efisien dalam berbagai skenario.

Implementasi algoritma ini menggunakan bahasa pemrograman C. Pemilihan C didasarkan pada beberapa alasan utama: (1) Efisiensi Kinerja, karena C memberikan kontrol langsung terhadap alokasi memori dan pengelolaan sumber daya, yang sangat penting dalam eksplorasi pohon permainan dengan skala besar; (2) Kompatibilitas, karena C dapat digunakan di hampir semua sistem operasi dan perangkat keras tanpa overhead yang besar; dan (3) Sederhana namun Fleksibel, memungkinkan pengembang untuk mendesain algoritma khusus sesuai kebutuhan tanpa banyak ketergantungan pada pustaka eksternal. Selain itu, penggunaan C mempermudah integrasi algoritma ini dengan perangkat lunak analitik atau simulasi lainnya yang umumnya mendukung standar C. Dengan pendekatan ini, algoritma yang diimplementasikan dapat mencapai efisiensi tinggi dalam menangani kompleksitas permainan catur.

E. Eksperimen dan Simulasi

Pada bagian ini, eksperimen dan simulasi dirancang untuk menguji efektivitas algoritma catur yang dikembangkan, khususnya dalam menentukan langkah optimal menggunakan pendekatan kombinatorika dan pohon permainan. Eksperimen dilakukan dengan mengukur tiga parameter utama: waktu komputasi, kedalaman pohon, dan hasil analisis game review oleh Chess.com. Permainan dilakukan melawan teman penulis sebagai subjek uji coba algoritma.

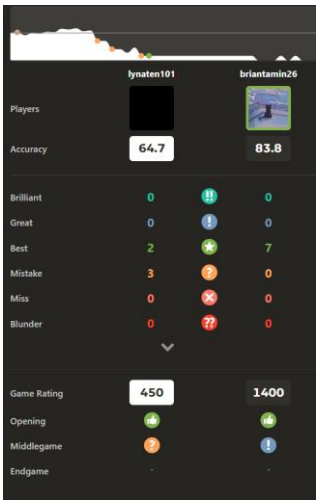
Eksperimen ini dilakukan dengan menggunakan implementasi algoritma dalam bahasa C, yang dipilih karena kecepatan dan efisiensi pengelolaan memori yang dibutuhkan dalam eksplorasi pohon permainan hingga kedalaman tertentu. Permainan simulasi dilakukan dalam skenario melawan teman dengan menggunakan langkah-langkah yang direkomendasikan oleh algoritma untuk setiap giliran.

Parameter yang diuji berupa waktu komputasi, kedalaman pohon, dan hasil *game review* oleh *Chess.com*. Waktu Komputasi diukur untuk mengevaluasi efisiensi algoritma pada kedalaman eksplorasi yang berbeda. Kedalaman Pohon: dievaluasi pada kedalaman 1 dan 2 untuk melihat pengaruh kedalaman terhadap kualitas langkah optimal. Permainan yang dihasilkan juga dianalisis menggunakan fitur game review Chess.com, dengan metrik seperti akurasi, kesalahan, blunder, dan evaluasi akhir permainan.

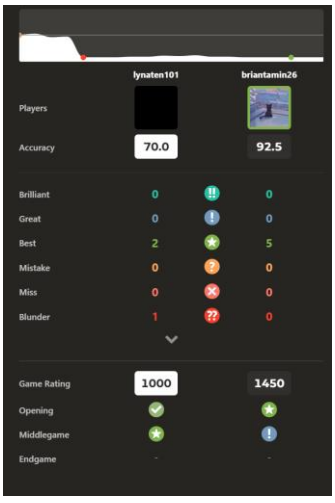
Pada proses generasi langkah, algoritma dimulai dengan membangun pohon permainan dari posisi awal (root node). Semua langkah legal dari posisi awal dihasilkan menggunakan fungsi seperti pawn_moves, queen_moves, combination, dan lainnya. Setiap langkah membentuk simpul baru dalam pohon permainan. Nilai evaluasi dari setiap simpul dihitung dengan algoritma evaluasi (main_evaluation) untuk menentukan langkah optimal. Permainan Simulasi Simulasi permainan dilakukan dalam melawan teman menggunakan langkah-langkah yang direkomendasikan algoritma. Setiap langkah dievaluasi berdasarkan akurasi permainan menggunakan fitur game review di Chess.com. Analisis Akurasi Permainan melawan teman dianalisis menggunakan Chess.com game review, yang memberikan metrik seperti akurasi (%) yang mencerminkan seberapa dekat langkah yang diambil dengan langkah optimal menurut mesin catur. Akurasi permainan dihitung sebagai parameter utama untuk menilai efektivitas algoritma dalam mendukung keputusan pemain. Terdapat juga jumlah kesalahan (*mistakes, misses, blunders*). Metrik ini mencerminkan kualitas langkah selama permainan berlangsung. Statistik tersebut digunakan untuk menganalisis dampak langkah yang direkomendasikan algoritma terhadap permainan nyata. Evaluasi akhir (*game rating*) menunjukkan kualitas keseluruhan permainan dari pembukaan hingga akhir.

IV. HASIL DAN ANALISIS

Berikut ini merupakan hasil eksperimen dan simulasi melawan teman saya dalam permainan catur menggunakan algoritma yang telah dibuat melalui *Chess.com*.



Gambar 2. Hasil game review oleh chess.com menggunakan kedalaman pohon 1.



Gambar 3. Hasil game review oleh chess.com menggunakan kedalaman pohon 2.

Di bawah ini merupakan tabel hasil eksperimen yang mengukur waktu komputasi berdasarkan kedalaman pohon.

Kedalaman Pohon	Waktu Komputasi (Detik)
1	0.5
2	14

Tabel 1. Perbandingan waktu komputasi dengan dua kedalaman pohon yang berbeda..

Dari hasil eksperimen, terlihat bahwa penambahan kedalaman pohon memengaruhi akurasi dan game rating secara positif. Pada kedalaman pohon 1, akurasi pemain sebesar 64.7%, sementara pada kedalaman pohon 2 meningkat menjadi 70.0%. Hal serupa terjadi pada jumlah kesalahan (mistake), yang menurun dari 3 kesalahan pada kedalaman 1 menjadi 0 kesalahan pada kedalaman 2. Game rating pun mengalami peningkatan signifikan, dari 450 menjadi 1000.

Penambahan kedalaman pohon memungkinkan algoritma mengevaluasi lebih banyak langkah legal, sehingga langkah yang diambil menjadi lebih optimal. Hal ini mencerminkan pentingnya eksplorasi yang lebih dalam dalam pengambilan keputusan permainan catur.

Namun, meskipun kedalaman pohon memberikan hasil yang lebih baik, kedua kedalaman ini masih belum cukup optimal. Hal ini terlihat dari kesalahan algoritma dalam beberapa skenario, seperti pemilihan langkah yang membuat raja berada dalam posisi terkena skak pada kedalaman pohon 1. Hal ini menunjukkan bahwa evaluasi langkah belum cukup mendalam untuk menghindari kesalahan-kesalahan kritis.

Salah satu tantangan signifikan dalam penambahan kedalaman pohon adalah peningkatan waktu komputasi. Hasil eksperimen menunjukkan bahwa waktu komputasi meningkat secara drastis dari 0.5 detik pada kedalaman 1 menjadi 14 detik pada kedalaman 2. Hal ini menunjukkan sifat eksponensial dalam eksplorasi pohon permainan catur, di mana setiap penambahan kedalaman memerlukan kalkulasi yang jauh lebih kompleks.

Implikasi ini sangat penting dalam konteks permainan cepat, di mana waktu adalah faktor krusial. Algoritma dengan kedalaman lebih tinggi memberikan hasil yang lebih akurat, tetapi membutuhkan optimisasi lebih lanjut agar tetap efisien secara waktu.

Meskipun hasil eksperimen menunjukkan peningkatan kinerja dengan penambahan kedalaman, algoritma masih memiliki beberapa kelemahan. Algoritma memilih langkah yang menyebabkan raja berada dalam posisi skak. Hal ini menunjukkan bahwa evaluasi langkah legal belum sepenuhnya akurat. Saat menjalankan algoritma pada kedalaman pohon 2, program terkadang berhenti tiba-tiba. Hal ini mengindikasikan adanya bug atau permasalahan manajemen memori pada implementasi algoritma.

V. Kesimpulan

Pendekatan pohon permainan dan kombinatorika terbukti efektif dalam menganalisis langkah optimal, di mana pohon permainan memungkinkan eksplorasi langkah-langkah legal berdasarkan kedalaman tertentu, dan kombinatorika membantu menghasilkan semua kemungkinan langkah legal dalam posisi tertentu. Berdasarkan hasil eksperimen, peningkatan kedalaman pohon dari 1 menjadi 2 menunjukkan peningkatan akurasi, penurunan jumlah kesalahan, dan kenaikan game rating, meskipun membawa konsekuensi berupa peningkatan waktu komputasi secara signifikan.

Eksperimen ini juga mengungkapkan beberapa kelemahan algoritma, seperti pemilihan langkah tidak legal (misalnya langkah yang menyebabkan raja berada dalam posisi skak) serta crash program pada eksplorasi kedalaman 2. Penggunaan bahasa C dalam implementasi algoritma memberikan keunggulan dalam kecepatan komputasi dan pengelolaan memori manual, namun tantangan utama tetap terletak pada eksplorasi pohon permainan yang kompleks. Untuk meningkatkan efisiensi algoritma, diperlukan optimisasi lebih lanjut, seperti penerapan teknik pruning (misalnya Alpha-Beta Pruning) dan pengelolaan memori yang lebih baik.

Hasil simulasi dalam permainan nyata melawan teman penulis menunjukkan bahwa algoritma ini mampu meningkatkan akurasi langkah, meskipun belum sepenuhnya optimal dalam mencegah kesalahan kritis. Selain itu, evaluasi dari platform Chess.com memperlihatkan adanya peningkatan performa algoritma berdasarkan kedalaman pohon, namun hasil ini masih memerlukan pengujian pada kedalaman yang lebih tinggi untuk mendapatkan analisis yang lebih komprehensif. Secara keseluruhan, penelitian ini menunjukkan potensi besar pendekatan pohon dan kombinatorika dalam menganalisis langkah optimal dalam catur, dengan catatan bahwa optimisasi dan pengembangan lebih lanjut tetap diperlukan untuk mencapai hasil yang lebih baik.

VI. Lampiran

- Video penjelasan makalah: https://youtu.be/oSBpVp_ZGqY
- Review *Chess.com* dengan pohon kedalaman 1: https://www.chess.com/analysis/game/live/130086617549?tab=revie
- Review *Chess.com* dengan pohon kedalaman 2: https://www.chess.com/analysis/game/live/130085418963?tab=review
- Repository github proyek: lynaten/chess

References

[1] Hxim. Stockfish Evaluation Guide. [Online]. Available: https://hxim.github.io/Stockfish-Evaluation-Guide/ [Accessed: 25 December 2024].

[2] Chessify. What is Depth in Chess: Different Depths for Stockfish and LCZero. [Online]. Available: https://chessify.me/blog/what-is-depth-in-chess-different-depths-for-stockfish-and-lczero [Accessed: 25 December 2024].

[3] The Chess World. Rules of Chess. [Online]. Available: https://thechessworld.com/basic-chess-rules/rules-of-chess/ [Accessed: 25 December 2024].

[4] Chessify. Chess Engine Evaluation. [Online]. Available: https://chessify.me/blog/chess-engine-evaluation [Accessed: 25 December 2024].

[5] Rinaldi Munir. Pohon (Bagian 1). [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf [Accessed: 25 December 2024].

[6] Rinaldi Munir. Pohon (Bagian 2). [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/24-Pohon-Bag2-2024.pdf [Accessed: 25 December 2024].

[7] Rinaldi Munir. Kombinatorika (Bagian 1). [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/18-Kombinatorika-Bagian1-2024.pdf [Accessed: 25 December 2024].

[8] Rinaldi Munir. Kombinatorika (Bagian 2). [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/19-Kombinatorika-Bagian2-2024.pdf [Accessed: 25 December 2024].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Januari 2025

Nathanael Rachmat - 13523142