

A Checkers Game based on Negamax Algorithm with Alpha Beta Pruning

¹Fatmasari, ²Rizky Tahara Shita, ³Lauw Li Hin

¹Program Studi Sistem Informasi, STMIK Antar Bangsa

^{2,3}Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

Email: ¹fsarie@gmail.com, ²rizky.taharashita@budiluhur.ac.id, ³lihin@budiluhur.ac.id

Abstract

Permainan Checkers merupakan salah satu permainan strategi dengan basis permainan papan (board game) dengan ukuran 8x8, dimainkan oleh dua pemain, dengan visualisasi kepingan bulatan, berjalan diagonal dan sebagai tujuannya, menghabiskan kepingan lawan. Walaupun Checkers terlihat seperti permainan logika sederhana jika dibandingkan dengan permainan strategi papan lainnya seperti catur, namun tetap memerlukan strategi logika. Permainan ini memang tidak terlalu dikenal dan tidak sering dimainkan di Indonesia, oleh karena itu harus dikemas dan dikembangkan dengan menarik agar lebih disukai. Rancangan aplikasi permainan ini dibuat pada intinya adalah mengasah logika murid – murid di SD Islam Al-Barkah. Jika sejak dini kemampuan logika juga diasah, maka bisa meningkatkan cara berfikir yang berujung pada peningkatan mutu. Algoritma yang digunakan dalam pembuatan permainan Checkers ini adalah algoritma Negamax With Alpha Beta Pruning sebagai algoritma pencarian langkah terbaik; dimana objektif dari algoritma ini adalah mencari masing – masing score dari node value yang dimiliki lalu dioptimasi dengan alpha beta Pruning untuk mempersingkat pohon pencarian dan menghasilkan bestscore sebagai keputusan langkah terbaik.

Keywords: game, negamax, algorithm

1. INTRODUCTION

Permainan digital yang memanfaatkan teknologi dapat memberikan hal positif maupun negatif; hal ini memang bergantung pada bagaimana pemain mengambil sudut pandang dalam memainkan permainan digital. Permainan digital yang memanfaatkan teknologi yang dapat membangun kreatifitas antara lain adalah permainan yang dapat membantu mengasah logika pemain sehingga dapat membangkitkan ide atau gagasan baru dari pemain tersebut. Tidak terlepas dari penerapan algoritma yang digunakan dalam membangun permainan digital ini agar pemain dapat mengasah logika; dan beberapa contoh permainan yang seperti ini antara lain adalah permainan catur digital, checkers, othello maupun tac-tic-toe. Dikarenakan permainan ini lebih banyak menggunakan pemikiran sebelum dapat melakukan aksi / langkah yang akan diambil, maka jenis permainan ini pun perlu dikemas dalam bentuk yang menyenangkan agar pemain dapat terus bermain dan lebih mengasah logika pemain tersebut. Mengasah logika seseorang sebaiknya dapat dimulai sejak dini; oleh karena itu adalah SD Islam Al-Barkah adalah salah satu SD Islam yang memiliki aktivitas harian dalam proses belajar – mengajar dapat menyisipkan pengasahan logika ini pada murid – murid melalui permainan digital yang dapat dilakukan pada saat matapelajaran komputer. Oleh karena itu, permainan digital dapat disisipkan pada saat penyampaian materi matapelajaran; sehingga murid - murid tidak hanya fokus pada materi apa

yang diterimanya saja, tetapi juga dapat mengasah logika melalui permainan digital yang menyenangkan. Permainan digital yang berupa permainan papan (*board game*) adalah permainan yang banyak memanfaatkan algoritma dalam mengatur langkah yang akan diambil oleh komputer yang berperan sebagai lawan dari pemain, sehingga diperlukan sebuah algoritma yang dapat membuat komputer menjadi “pintar” dalam bertanding melawan pengguna akan tetapi tetap dapat menyenangkan dalam melakukan permainan tersebut dan dapat membantu para murid dalam mengasah logika melalui permainan komputer. Salah satu algoritma yang dapat diterapkan dalam permainan digital adalah Negamax with Alpha Beta Pruning, sehingga dapat membuat komputer memiliki kecerdasan buatan sederhana dalam melawan pemain dan algoritma ini juga yang secara tidak langsung akan mendidik murid – murid dalam mengasah logika sejak dini melalui permainan digital dalam bentuk permainan papan (*board game*).

1.1. Masalah

Dengan melihat pada kondisi yang ada di SD Islam Al-Barkah, maka dapat dibuat rumusan masalah yang ada yaitu berada pada cara belajar – mengajar yang dilakukan hanya berdasarkan kurikulum yang ada saja; sekolah juga menginginkan agar memiliki kegiatan yang dapat mengasah logika berfikir murid – murid sejak dini dengan cara yang menyenangkan, dimana salah satunya adalah melalui permainan yang dapat mendidik secara tidak langsung.

1.2. Tujuan

Melihat dari permasalahan yang ada, maka tujuan dari dikembangkannya permainan digital ini adalah agar para murid dapat meningkatkan pola logika berfikir sejak dini dengan cara yang menyenangkan; dimana para murid yang bermain akan dapat melawan komputer yang dibekali dengan algoritma Negamax with Alpha Beta Pruning.

2. METHODS

2.1. Artificial Intelligence

Tahun 1950-an adalah awal mula dimana program berbasis *Artificial Intelligence* (AI) yang digunakan untuk menjalankan mesin Ferranti Mark pada Universitas Manchester; dimana program ini adalah sebuah permainan naskah yang ditulis oleh Christopher Strachey yang berikutnya hadir lagi permainan catur dengan memanfaatkan Artificial Intelligence yang dibuat oleh Dietrich Prinz. Adapun istilah Artificial Intelligence ini dikenalkan oleh professor John McCarthy dari Massachusetts Institute of Technology pada tahun 1956. Dan sejak saat itu, mulailah bermunculan program komputer lainnya yang memanfaatkan *Artificial Intelligence* yang tidak saja hanya dalam bentuk permainan komputer, tetapi berkembang implementasinya pada robot untuk keperluan industri.

Pengertian dari *Artificial Intelligence* itu sendiri adalah sebuah sistem kecerdasan buatan yang menggunakan logika tertentu yang diimplementasikan pada komputer, sehingga sebuah aksi / kegiatan

dapat dijalankan sendiri oleh komputer seperti layaknya yang dilakukan oleh manusia. *Artificial Intelligence* juga merupakan bagian dari ilmu komputer yang dapat membuat mesin (dengan menggunakan komputer) agar dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. [1]

2.2. Algoritma

Merupakan kumpulan perintah dalam bentuk program komputer untuk dapat menyelesaikan permasalahan dimana perintah ini diterjemahkan secara bertahap; dimana sebuah permasalahan dapat memiliki sebuah kondisi tertentu yang harus dipenuhi oleh algoritma untuk dapat diselesaikan. Klasifikasi algoritma dan metode yang digunakan dibagi menjadi:

- *Divide and Conquer*
adalah algoritma untuk membagi permasalahan yang besar kedalam permasalahan yang lebih kecil agar dapat lebih mudah diselesaikan.
- *Dynamic Programming*
merupakan jenis algoritma yang sama seperti *Divide and Conquer*, hanya saja lebih difokuskan pada pemecahan permasalahan yang bertumpuk (sub-struktur).
- *Greedy*
metody greedy mirip dengan *Dynamic Programming*, dimana hanya jawaban dari sub-struktur tersebut tidak perlu diketahui tahapannya dan hanya memilih yang terbaik.

2.3. Negamax

Menggunakan teknik pencarian *Depth First Search* adalah salah satu poin yang dilakukan pada algoritma Negamax; dimana proses yang dilakukan adalah melakukan pengecekan untuk setiap kemungkinan yang ada maka akan didapatkan pohon keputusan yang berisi kemungkinan – kemungkinan yang dapat diambil. Pada algoritma Negamax diimplementasikan juga kemungkinan negatif dari keputusan yang ada dan evaluasi pada algoritma Negamax menggunakan evaluasi status, sehingga diperlukan untuk melakukan asumsi terhadap langkah apa yang akan diambil dalam menentukan langkah terbaik. Pada algoritma Negamax ini juga akan dilakukan evaluasi berdasarkan dari nilai balik terhadap langkah dari lawan; dimana *Pseudo code* dari algoritma Negamax dapat dilihat sebagai berikut: [2]

```
1. function negamax(node, depth, color)
2.   if depth = 0 or node is a terminal node
3.     return color * the heuristic value of node
4.     bestValue := -∞
5.
6.   foreach child of node
7.     val := -negamax(child, depth-1, -color)
```

```

8.         bestValue := max( bestValue, val)
9. return bestValue
    
```

```

1. // initial call for player A's root node
2. rootNodeValue := negamax (rootNode, dept, 1)
    
```

```

1. // initial call for player B's root node
2. rootNodeValue := -negamax (rootNode, dept, -1)
    
```

Dapat dilihat bahwa dalam mendapatkan nilai dari variabel *bestValue* untuk satu pemain dilakukan dari sudut pandang pemain lainnya yang dapat saja pemain tersebut mendapatkan nilai max maupun nilai min. Negamax akan melakukan penelusuran pada keseluruhan node yang ada pada pohon pencarian dari awal sampai akhir untuk mendapatkan score yang memiliki konsekuensi lebih lamanya waktu dalam memproses hal ini. Seperti juga dengan Minimax, yang mana jika nilai variabel *d* (kedalaman maksimum – *depth*) dan variabel *b* (langkah – branch) dilakukan pada setiap node, maka time complexity dari algoritma adalah $O(b^d)$; berikut ini adalah gambar dari pohon algoritma Negamax: [3]

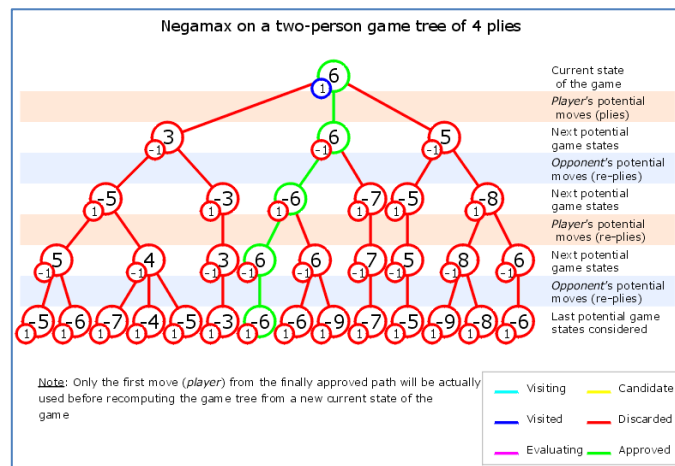


Figure 1 Pohon Negamax

2.4. Alpha Beta Pruning

Salah satu kelemahan dari algoritma Negamax adalah waktu dalam melakukan proses, sehingga dibutuhkan cara agar dapat mempersingkat proses dari kemungkinan - kemungkinan yang akan dianalisa dengan tujuan yang sama tetapi memiliki waktu proses yang lebih singkat.

Salah satu cara optimasi yang dapat dilakukan adalah dengan menerapkan Alpha Beta Pruning dalam mengurangi jumlah node yang akan dievaluasi. Algoritma Alpha Beta Pruning akan berhenti melakukan evaluasi langkah pada saat ada minimal satu langkah yang nilainya lebih buruk

dibandingkan dengan langkah yang dievaluasi sebelumnya. Hal ini akan membuat langkah tersebut menjadi tidak perlu dilakukan evaluasi, sehingga algoritma utama dari Negamax tetap (tidak berubah).

Pada Alpha Beta Pruning; variabel alpha digunakan untuk menunjukkan nilai bawah dari node max, sedangkan variabel beta menunjukkan batas atas node min. Saat node min dilakukan evaluasi, maka proses akan berhenti jika telah terdapat *child node* yang sudah mempunyai nilai lebih kecil dibandingkan dengan nilai batas bawah. Sebaliknya pada saat node max, evaluasi akan berhenti jika telah didapat *child node* yang mempunyai nilai lebih besar dibandingkan dengan nilai batas atas.

Pada pohon pencarian; nilai alpha diisi dengan nilai $-\infty$ dan nilai beta diisi dengan nilai $+\infty$. Node yang melakukan maksimasi akan melakukan perbaikan pada nilai alpha dari *child node* -nya dan node yang melakukan minimasi akan melakukan perbaikan nilai beta dari nilai *child node* -nya dimana kesimpulannya adalah jika $\alpha > \beta$ maka proses evaluasi akan distop. [1].

Jika pada setiap *depth* terdapat kondisi *best case*, maka langkah terbaik selalu didapatkan. Hal ini dilakukan agar time complexity menjadi lebih kecil atau dengan kesimpulan. *Pseudo code* Negamax with Alpha Beta Pruning adalah sebagai berikut: [4]

```

1. int AlphaBeta (pos, depth, alpha, beta) {
2.     if (depth == 0) return
3.
4.     evaluate(pos);
5.     best = -INFINITY;
6.     succ = Successors(pos);
7.     while (not Empty(succ) && best < beta) {
8.         pos = RemoveOne(succ);
9.         if (best > alpha) alpha = best;
10.        value = -AlphaBeta(pos,depth-1, -beta, -alpha);
11.        if (value > best) best = value;
12.    }
13.
14.    return best;
15.}

```

Ketergantungan optimasi Alpha Beta terletak pada urutan simpul yang sedang dievaluasi. Seperti halnya Negamax, algoritma juga akan membuka node yang sama; akan tetapi jika nilai Beta lebih kecil dari Alpha, maka dapat disimpulkan bahwa node tersebut adalah hasil terbaik dan tidak perlu dilakukan evaluasi lebih lanjut. Berikut ini adalah gambar pohon algoritma Alpha Beta Pruning: [3]

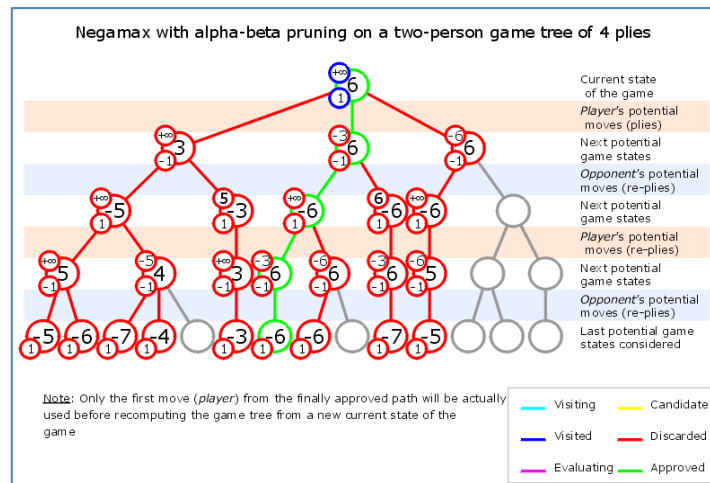


Figure 2 Pohon Alpha Beta

Terlihat bahwa evaluasi dari pohon pencarian tidak dilanjutkan pada beberapa posisi yang terlihat kosong nilainya dimana pada node pertama adalah node dengan nilai kosong yang disebut dengan alpha cutoff (dimana pada posisi node tersebut didapatkan nilai yang lebih kecil); sehingga penelusuran tidak perlu lagi dilanjutkan. Saat kondisi kedua, didapatkan hasil penelusuran dengan nilai yang lebih kecil dan penelusuran pun dihentikan. Pada saat kondisi diposisi akhir; dimana didapatkan nilai sudah pada batas paling rendah dari Alpha, sehingga tidak perlu lagi dilanjutkan proses penelusuran. Dapat diambil kesimpulan bahwa dengan menggunakan Alpha Beta, proses pencarian menjadi lebih singkat yang disebabkan tidak semua node dilakukan evaluasi dan terlihat juga bahwa proses ini tidak mengubah nilai dari hasil pencarian.

2.5. Permainan Checkers

Merupakan sebuah permainan papan (*board game*) dengan ukuran 8 x 8 dimana menggunakan kepingan bulat sebagai bidak yang berjalan secara diagonal dengan cara melompati lawan untuk memakan keping lawan. Dibutuhkan strategi dan logika yang baik dalam mengambil langkah dimana yang akan menang adalah yang berhasil menghabiskan kepingan lawan. Dua pemain dibutuhkan dalam bermain permainan ini, dimana untuk aturan langkah tiap keping telah ditetapkan oleh *Word Checkers Draughts Federation* (WCDF).

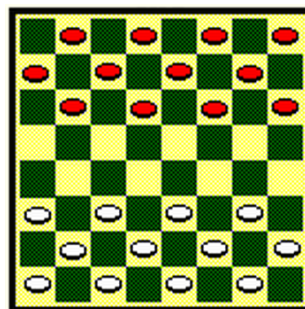


Figure 3 Permainan Checkers

3. RESULTS AND DISCUSSION

3.1. Analisa dan Penyelesaian Masalah

Dengan melihat kondisi yang ada pada SD Islam Al-Barkah dimana proses belajar mengajar dilakukan hanya berdasarkan dari kurikulum yang ada dan diperlukannya sebuah cara agar murid SD Islam Al-Barkah dapat mengasah logika berfikir para murid dengan cara yang menyenangkan, maka salah satu cara yang dapat direalisasikan adalah dengan mengembangkan permainan komputer jenis papan (*board game*). Dimana pada permainan komputer yang dikembangkan ini, nantinya akan dapat disisipkan pada matapelajaran komputer dan murid – murid dapat melatih logika berfikir sejak dini dengan melawan komputer yang memiliki kecerdasan buatan sederhana yang mana pada permainan komputer tersebut disematkan algoritma dan logika untuk dapat menandingi pemain. Jenis permainan komputer yang dikembangkan adalah permainan papan (*board game*) yang dikenal dengan nama Checkers dan agar komputer dapat memiliki kecerdasan buatan sederhana, maka algoritma yang disematkan adalah Algoritma Negamax with Alpha Beta Prunning. Penggunaan algoritma Negamax ini disematkan agar komputer dapat melakukan pengambilan langkah dari kemungkinan - kemungkinan yang ada dan untuk mempercepat proses dari pengambilan keputusan hasil dari algoritma Negamax, maka diterapkan konsep pemangkasan proses dengan menambahkan logika Alpha Beta Prunning dengan tetap tidak mengubah hasil dari algoritma utama nya.

3.2. Aturan permainan Checkers

Untuk dapat mengembangkan permainan checkers ini, maka perlu diketahui aturan permainan checkers itu sendiri; dimana nantinya aturan ini akan diterapkan sebagai landasan bagi komputer dalam mengambil langkah pada saat melawan dengan pemain. Aturan permainan checkers ini sudah ditetapkan secara baku melalui karya tulis yang dibuat oleh Jimloy pada tahun 1999 dengan judul “*The Standard Laws of Checkers*” dan dipublikasikan oleh badan WCDF (*World Checkers and Draughts Federation*). Aturan permainan checkers itu sendiri antara lain terdiri dari keping checkers normal yang mana memiliki warna yang berbeda untuk tiap pemain dan masing – masing mendapatkan 12 keping. Sebagai bidang dari keping checkers ini, maka terdapat papan permainan dengan ukuran 8 kolom x 8 baris yang memiliki warna berbeda dan diposisikan secara selang – seling. Adapun posisi pada saat permainan dimulai, memiliki aturan penempatan sebagai berikut:

Setiap pemain memiliki cara untuk melangkahkan kepingannya yang secara normal dilakukan diagonal menuju arah lawan; sedangkan untuk kepingan raja dapat berjalan 1 secara diagonal tanpa harus menuju kearah lawan.

Pada saat pemain akan memakan kepingan lawan, maka dilakukan dengan cara melompati (*jump*) kepingan tersebut; dimana jika pemain memiliki kesempatan untuk melakukan *jump*, maka pemain tersebut harus melakukannya dan hal ini dikenal dengan istilah *Force Move to Jump* [5]. Dan hal ini juga untuk dapat memakan kepingan lawan lebih dari satu dalam putaran langkah sang pemain.



Figure 4 Force Move to Jump

Jika terdapat langkah yang dilakukan terdapat dalam ketentuan tersebut, maka disebut dengan *Legal Move*; sedangkan cara melangkah yang tidak sesuai dengan aturan disebut dengan *Illegal Move*. Sehingga dengan memahami aturan dari permainan, maka pengembangan permainan komputer dapat disesuaikan dan diterapkan aturannya dan tidak saja dengan penyematan algoritma Negamax with Alpha Beta Prunning. Agar kepingan dapat berubah menjadi kepingan raja, maka kepingan normal harus ditempatkan pada posisi paling atas dari arah pemain. Dengan adanya kepingan normal yang sudah berada pada posisi ini, maka kepingan normal dapat berubah menjadi kepingan raja dengan aturan jalan yang sudah dibahas sebelumnya. Untuk menentukan pemenang permainan, maka ada beberapa kondisi yang perlu diperhatikan; yaitu pada saat salah satu pemain berhasil menghabiskan kepingan lawannya atau pada saat kepingan tidak dapat melakukan langkah.

3.3. Penerapan Algoritma Negamax with Alpha Beta Prunning

Utamanya dari algoritma Negamax adalah menentukan pengambilan keputusan terbaik dari kemungkinan yang ada; akan tetapi proses yang dimiliki oleh algoritma ini cukup memakan waktu, sehingga dibutuhkan sebuah cara dan logika pendamping agar proses utama dalam pengambilan keputusan tetap dapat dilakukan tetapi memiliki waktu proses yang lebih singkat. Penerapan Alpha Beta Prunning ini dapat membantu pemangkasan proses Algoritma Negamax tanpa mengubah hasil akhirnya. Secara singkat, tahapan penerapannya adalah sebagai berikut:

- Dapatkan nilai Alpha dan Beta; dimana Alpha adalah nilai maksimum sementara dan Beta adalah nilai minimum yang ditentukan.
- Saat posisi pointer berada pada node maksimum dan sebelum dilakukan evaluasi *child node*, maka dilakukan perbandingan terlebih dahulu terhadap nilai yang didapat dengan nilai Beta. Jika lebih besar, maka tidak perlu dilakukan pencarian terhadap *child node* dan nilai yang didapatkan adalah nilai maksimum.
- Saat posisi pointer berada pada node minimum dan sebelum dilakukannya evaluasi *child node*, dilakukan perbandingan antara nilai yang didapat dengan nilai Alpha. Jika nilai lebih kecil, maka tidak perlu dilanjutkan untuk melakukan pencarian untuk node tersebut dan nilai yang dikembalikan adalah nilai minimum.

3.4. Flowchart Permainan

Permainan komputer ini pada awal dijalankan akan menampilkan layar konfigurasi yang dapat disesuaikan dengan kebutuhan pemain sebelum pemain memmainkannya. Adapun *flowchart* permainan dapat dilihat pada gambar berikut ini:

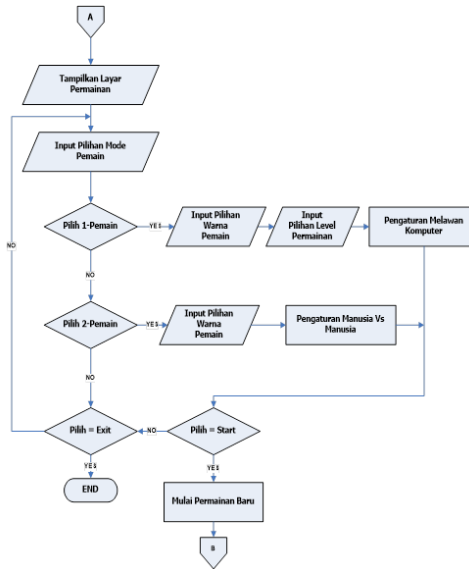


Figure 5 Flowchart Permainan

Sedangkan, pada saat pemain sudah memutuskan untuk bermain; maka *flowchart* utama permainan tampak pada gambar berikut ini:

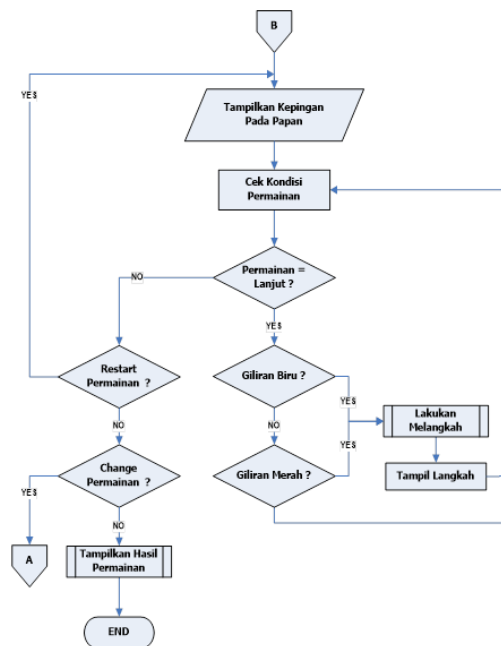


Figure 6 Flowchart Utama Permainan

3.5. Flowchart Negamax with Alpha Beta Pruning

Penerapan dari Algoritma Negamax with Alpha Beta Pruning yang merupakan inti utama dari kecerdasan tiruan sang komputer dalam menentukan langkah terbaik dapat dilihat pada gambar berikut:

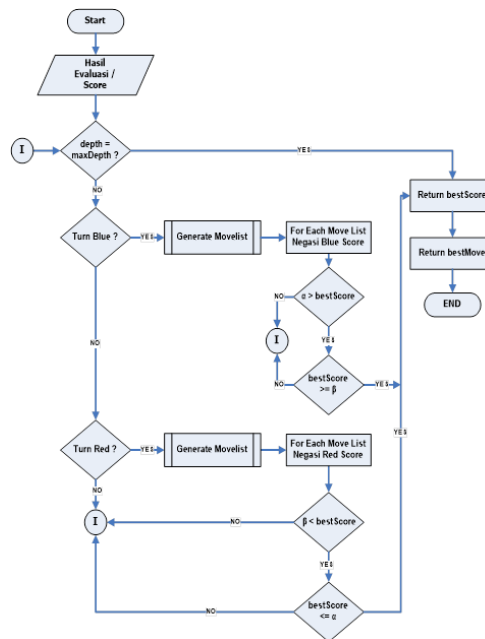


Figure 7 Flowchart Negamax with Alpha Beta Pruning

Pada *flowchart* ini menjelaskan proses dari melakukan evaluasi yang sedang berlangsung dan pembatasan kedalaman dalam melakukan pencarian node untuk mendapatkan nilai dari variabel *bestScore* yang berakhir pada penentuan langkah yang akan dilakukan oleh komputer.

3.6. Implementasi Permainan

Pertama kali permainan komputer ini dijalankan, maka akan menampilkan layar pembuka dan akan menampilkan konfigurasi permainan terlebih dahulu. Setelah dilakukan konfigurasi oleh pemain, maka permainan checkers akan dimulai sesaat setelah pemain memilih tombol *Start*.

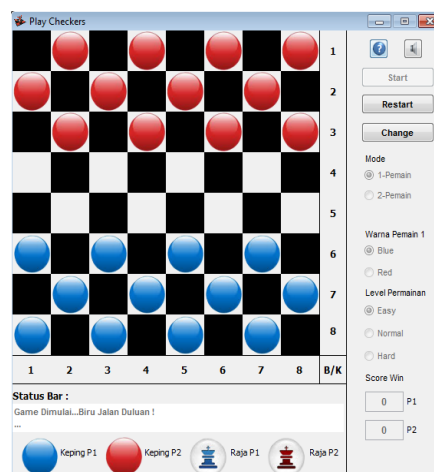


Figure 8 Tampilan Permainan Checkers

Pemain akan mendapatkan tampilan seperti pada gambar untuk langsung dapat bermain. Adapun fitur yang tersedia untuk dapat dilakukan konfigurasi oleh pemain antara lain adalah fitur suara, mode pemain (1 pemain melawan komputer atau 2 pemain yang masing – masing melawan manusia), warna kepingan, level permainan serta riwayat nilai (*score*) dari permainan yang sudah dimainkan sebelumnya.

3.7. Pengujian Aplikasi

Untuk dapat mengukur kecerdasan buatan yang dimiliki oleh komputer pada permainan checkers ini, maka dilakukan pengujian kepada beberapa murid – murid SD Islam Al-Barkah. Berikut ini adalah hasil dari pengujian yang telah dilakukan untuk 10 kali permainan:

Table 1 Pengujian oleh Murid 1

Level	Menang	Kalah
Easy	6	4
Normal	1	9
Hard	0	10

Table 2 Pengujian oleh Murid 2

Level	Menang	Kalah
Easy	8	2
Normal	2	8
Hard	0	10

3.8. Evaluasi

Agar dapat menganalisa baik atau tidaknya terhadap permainan komputer yang dikembangkan, maka dari analisa, pengembangan serta pengujian yang telah dilakukan; maka berikut ini adalah evaluasi yang dapat disimpulkan:

- Kelebihan
 - Terdapat fitur konfigurasi permainan sebelum pemain bermain checkers.
 - Terdapat fitur riwayat langkah yang sudah dijalankan.
 - Memiliki algoritma kecerdasan buatan sederhana dengan memanfaatkan Algoritma Negamax with Alpha Beta Pruning.
- Kekurangan
 - Game masih berbasis desktop, tapi tidak menutup kemungkinan untuk dilakukan migrasi ke platform mobile.

- Belum dapat dimainkan secara jaringan / online; sehingga jika dimainkan oleh 2 pemain manusia (bukan melawan komputer), harus dilakukan secara bergantian pada saat melakukan langkah.
- Belum adanya fasilitas untuk menyimpan maupun membuka ulang kondisi permainan.
- Belum terdapat fitur untuk undo terhadap langkah yang sudah dilakukan

4. CONCLUSION

Dari hasil analisa kebutuhan, pembahasan permasalahan dan pengujian permainan yang dikembangkan; maka dapat disimpulkan, bahwa permainan papan (*board game*) yang dikembangkan dapat dijadikan salah satu kegiatan tambahan bagi murid – murid di SD Islam Al-Barkah untuk melatih perkembangan pikir dan mengasah logika sejak dini, sehingga dapat membuat kegiatan belajar mengajar menjadi lebih menyenangkan. Penerapan algoritma Negamax with Alpha Beta Pruning dapat melakukan pencarian langkah terbaik, sehingga membuat komputer dapat memiliki kecerdasan dalam memberikan perlawanan. Permainan juga difasilitasi dengan fitur yang cukup lengkap untuk sebuah permainan sederhana, sehingga pengguna dapat melakukan beberapa konfigurasi sebelum permainan dimainkan.

REFERENCES

- [1] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*, 1st ed. Yogyakarta: Graha Ilmu, 2003.
- [2] D. M. Breuker, “Memory versus Search in Games,” SIKS, 1998.
- [3] Wikipedia, “Negamax,” *Wikipedia*, 2023. <https://en.wikipedia.org/wiki/Negamax> (accessed Mar. 27, 2023).
- [4] A. E. / I. T. B. Luman, “Pengaplikasian Pohon dalam Algoritma Sebuah Game Catur,” 2008, [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2008-2009/Makalah2008/Makalah0809-040.pdf>
- [5] J. Loy, “The Standard Laws of Checkers,” 2009. <http://www.jimloy.com/checkers/rules.htm> (accessed Nov. 23, 2013).