

Penerapan *Augmented reality* Berbasis *Minimax Algorithm* pada Game Papan Cerdas

Latius Hermawan¹, Maria Bellanar Ismiati²

^{1,2}Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Katolik Misi Charitas
Jl. Bangau No. 60, Palembang 30113, Sumatera Selatan, Indonesia
Email: ¹tiuz.hermawan@ukmc.ac.id, ²bella@ukmc.ac.id

Abstract. *Application of Augmented reality Based on Minimax-Alpha Beta Pruning Algorithm on Smart Board Games.* Augmented reality technology is growing very rapidly making game production more innovative and attractive. The implementation of this technology also has the potential for traditional board games which are starting to be replaced by computer-based digital games. The method used in the digital board is Minimax which is zero-sum based where one point of the opponent's victory will reduce the player's one point. This method underlies the way of thinking to get critical steps in several types of games being played. Minimax will result in a lower probability of defeat and increase the probability of winning. The results obtained are that Minimax which was developed with Alpha Beta Pruning to make opponents think like humans so that artificial intelligence in it is suitable to be applied. The test results also give a 63% win for the AI (Artificial Intelligence) used, so the game becomes challenging.

Keywords: *Game, augmented reality, Minimax, Board*

Abstrak. Teknologi *Augmented reality* yang berkembang sangat pesat membuat produksi *game* lebih inovatif dan atraktif. Implementasi teknologi tersebut juga berpotensi untuk permainan papan tradisional yang mulai tergantikan oleh permainan digital berbasis komputer. Metode yang digunakan dalam *digital board* adalah Minimax yang berbasis *zero-sum* dimana satu poin kemenangan lawan akan mengurangi satu poin pemain. Metode ini mendasari cara berfikir untuk mendapatkan langkah-langkah kritis dalam beberapa jenis *game* yang dimainkan. Minimax akan menghasilkan kemungkinan kekalahan yang sedikit dan memperbanyak kemungkinan kemenangan. Hasil yang didapatkan yaitu Minimax yang dikembangkan bersama Alpha Beta Pruning mampu membuat lawan berfikir layaknya manusia sehingga kecerdasan buatan didalamnya cocok untuk diterapkan. Hasil pengujian juga memberikan hasil 63% kemenangan bagi AI (*Artificial Intelligence*) yang digunakan, sehingga permainan menjadi menantang.

Kata Kunci: *Game, augmented reality, Minimax, papan*

1. Pendahuluan

Dalam beberapa waktu terakhir teknologi yang bernama *augmented reality* mulai berkembang sangat pesat. Perkembangan *augmented reality* memberikan pengaruh yang baik terhadap *game*. Teknologi *augmented reality* yang semakin berkembang akan membuat *game* yang telah dibuat menjadi lebih inovatif dan atraktif [1]. Teknologi yang dimiliki *augmented reality* dapat diterapkan pada *game* tradisional dan digital. Banyak sekali *game* tradisional yang ada di Indonesia serta cukup digemari, misalnya *game* berbasis papan. Permainan papan hingga saat ini masih banyak dimainkan oleh masyarakat dari anak-anak sampai orang dewasa karena cara bermainnya yang mudah dimengerti [2]. Permainan yang berbasis papan sudah lama dimainkan di Indonesia. Sesuai *survey* acak yang telah dilakukan, sebanyak 20 anak yang berumur 6-12 tahun masih sering memainkan permainan dengan papan, sedangkan 20 orang dewasa (17-30 tahun) juga masih memainkan permainan papan (catur, othello, dll) dan sisanya sebanyak 60 responden menyatakan sudah tidak memainkannya lagi. Salah satu penyebab permainan ini masih digunakan adalah alat yang dibutuhkan hanya berupa papan permainan dan bidak. Sayangnya, permainan tradisional yang berbasis papan ini mulai ditinggalkan karena tergantikan dengan alat digital seperti *smartphone*, konsol *game* dan komputer.

Banyak orang muda hingga dewasa mengenal aplikasi permainan yang ada dalam komputer karena sangat interaktif dan mudah untuk dimainkan. Aplikasi yang cukup mudah dan menyenangkan ini sering dipakai oleh para pemain untuk menghilangkan stress dan mengisi waktu kosong [3]. Penggunaan aplikasi *game* sering dilakukan sendiri karena tidak ada lawan untuk dimainkan (tanpa adanya *Artificial Intelligence*). Permainan dengan dua *player*, misalnya: Catur, *Tic Tac Toe*, Janggi, dan Othello. Permainan berbasis papan yang akan diteliti selanjutnya adalah Othello. Othello merupakan *game* meja yang cara memainkannya menggunakan papan dengan ukuran $n \times n$ segmen dan setiap pemain memiliki bidak (hitam atau putih). *Winner* akan ada jika saat *end game*, pemain memiliki jumlah bidak > 1 dibandingkan lawan. *Game* tradisional ini membutuhkan sebuah metode agar dapat dibuat dalam bentuk digital dan dapat dimainkan pada *smartphone* dengan teknologi *augmented reality*. Tujuannya, *game* berbasis papan dapat terus dimainkan oleh siapapun dan permainan ini akan tetap ada. Permainan berbasis papan ini juga membutuhkan algoritma agar dapat bermain layaknya manusia, sehingga akan membuatnya lebih menarik dan ada interaksi didalamnya.

Kecerdasan buatan pada *game* digital adalah salah satu bidang rekayasa perangkat lunak yang dicirikan sebagai AI (*Artificial Intelligence*) yang kerangka kerjanya berbasis perhitungan tertentu. Tujuan kerangka kerja tersebut secara keseluruhan ialah siap untuk berpikir seperti manusia [4]. Beberapa algoritma yang sering digunakan untuk melengkapi AI pada *game* ialah A*, Dijkstra, Minimax, Bruteforce, BFS (*Breadth First Search*), dan DFS (*Depth First Search*). Pada penelitian ini, algoritma yang digunakan adalah Minimax karena sifatnya yang rakus dan mencari keuntungan besar dan ditambahkan metode *Alpha Beta Pruning* yang telah disesuaikan untuk mengurangi simpul yang akan dikunjungi, sehingga proses penentuan keputusan diharapkan menjadi lebih cepat. Metode Minimax ini digunakan dalam *board game* (*digital*) dengan dua *player* serta berbasis *zero-sum* (1 poin kemenangan lawan akan mengurangi 1 poin pemain). Pada umumnya, ide dari perhitungan Minimax ini adalah untuk membatasi peluang kekalahan dan memperluas peluang kemenangan [5].

Karena persyaratan ini, penelitian ini bertujuan membuat aplikasi *game* terkomputerisasi yang dikemas dalam *Augmented Reality* agar permainan Othello dapat dimainkan dengan lebih menarik dan mengikuti perkembangan zaman. Dengan inovasi realitas yang diperluas, permainan meja tradisional ini bisa menjadi lebih menarik.

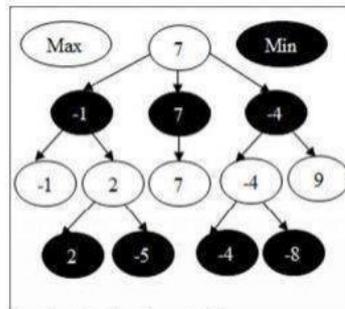
2. Tinjauan Pustaka

2.1. *Game*

Menurut [6] ada empat elemen dasar dari *game*: (1) mekanisme yang berisi aturan dalam sebuah *game*, (2) urutan kejadian dari keseluruhan *game*, (3) estetika dimana menjadi khas dari *game*, (4) komponen teknologi yang dapat menjadikan permainan melakukan aksi tertentu, membentuk inti *game*, dan posisi letak dari *game*. *Board Games* adalah permainan potongan di atas permukaan papan yang dapat dipindah / diletakkan sesuai alur cerita dari permainannya. Permainan dilakukan berdasarkan strategi, kesempatan, dan tujuan yang ingin dicapai pemain [3].

2.2. Algoritma Minimax – Alpha Beta Pruning

Algoritma Minimax adalah algoritma AI di permainan / di *zero-sum games* (Catur, Othello) [7]. Cara kerja Minimax ialah pengecekan dilakukan pada semua kemungkinan bahkan hingga akhir permainan. Hasilnya adalah pohon permainan untuk semua pengetahuan rute yang mungkin akan dilewati. Lalu komputer melakukan analisis semua pohon permainan. Pada setiap langkahnya, dipilih langkah yang paling baik. Hal tersebut agar lawan mendapatkan keuntungan minimum dan keuntungan maksimum sesuai dengan ciri khas dari algoritma Minimax [8]. Gambar 1 menunjukkan contoh Minimax.



Gambar 1. Contoh Minimax

Saat menentukan keputusan, perlu nilai dan bobot yang merupakan representasi kerugian / keuntungan dari pemain di tiap langkah pada permainan, sehingga memiliki hasil penentuan terbaik (keuntungan paling besar serta kerugian paling kecil) [5]. Minimax adalah metode yang sering digunakan dalam permainan berbasis AI. Pencarian seluruh kemungkinan tersebut akan membuat solusi yang akan membutuhkan waktu lama sehingga Minimax perlu disesuaikan dengan mengatur kemungkinan simpul mana yang akan dikunjungi. Metode *Alpha Beta Pruning* bekerja dengan jumlah simpul yang dikurangi untuk di-*explore*. Waktu yang diperlukan saat pencarian solusi berbasis pohon akan berkurang. Hal ini karena adanya pembatasan waktu ketika pohon permainan dievaluasi. Implementasinya memberi jalur yang optimal pada tiap kemungkinan solusi di pohon permainan saat dieksplorasi oleh Minimax [5].

2.3. Augmented reality

Menurut [10], *Augmented reality* (AR) merupakan sistem yang memiliki tujuan pengembangan teknologi dalam menggabungkan konten digital *real-time* dan dunia nyata. Pengguna mampu melihat dunia nyata yang ada terdapat penambahan objek virtual ke dunia nyata. Menurut Alan. B. Craig [11], AR merupakan ide dalam menghubungkan manusia serta komputer atau sebaliknya untuk mengimplementasikan *Augmented Reality* yang merupakan teknologi terkini. Dalam buku [12], dikenalkan istilah *Mixed Reality*. Deskripsinya terdapat kekurangan yang memisahkan lingkungan nyata serta virtual. Diantaranya, ada dua bagian penghubung yang punya ciri khas berbeda. Keduanya merupakan *Augmented reality* serta *virtuality*.

2.4. Markerless

Menurut Mario Vento [13], perkembangan metode *Augmented reality* (AR) adalah *Markerless Augmented reality*. Metodenya membuat pengguna tidak memakai *marker* dalam menampilkan objek-objek digital. Sekarang telah berkembang satu teknologi terbaru untuk teknik-teknik *Markerless tracking*.

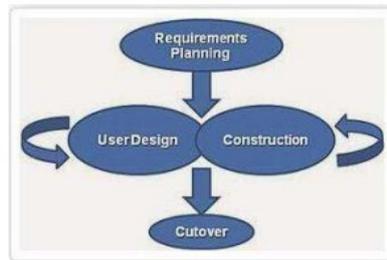
2.5. Vuforia

Menurut Mario Fernando [14], Vuforia merupakan *tools Augmented reality*, dalam bentuk *Software Development Kit* (SDK) dan dipakai di *smartphone*. SDK pada Vuforia memakai *computer vision* agar mampu mengenal serta melacak target dengan *real time*. Karenanya, pengembang bisa membuat objek virtual terlihat dari kamera di *smartphone*. Selanjutnya objek mencari letak orientasi serta gambar dengan *real time* yang akhirnya membuat pengguna melihat objeknya di *smartphone* dan seakan-akan virtual objek ada di dunia nyata. SDK dari Vuforia memiliki beberapa jenis target (2D dan 3D) dan *multi target*, seperti bentuk silinder.

2.6. Rapid Application Development

Rapid Application Development (RAD), salah satu model yang ada dalam pembuatan *software* yang termasuk teknik *incremental*. Fokusnya pada pembangunan *software* yang cepat.

Singkatnya waktu adalah syarat utama untuk membuat perangkat lunak yang menggunakan model RAD. RAD menerapkan proses iteratif saat pengembangan sistem dan saat menentukan *requirement* hingga perangkat lunak selesai dibuat [15]. Gambar 2 menunjukkan proses dari *Rapid Application Development*.



Gambar 2. Proses *Rapid Application Development*

3. Metodologi Penelitian

3.1. Proses Penelitian

Research ini membutuhkan waktu satu semester untuk menyelesaikannya. Secara keseluruhan, proses penelitian ini telah melewati beberapa tahap yang telah dilakukan seperti:

a. Tahap I

Analisis dan identifikasi masalah dilakukan secara mendalam pada tahap ini. Daftar identifikasi masalah dan metode yang digunakan akan menjadi inputan awal. Proses penelitian meliputi analisis terhadap masalah, algoritma, serta tools yang cocok dipakai untuk membuat aplikasi tersebut.

b. Tahap II

Dalam proses ini, hasilnya yaitu dibuatnya suatu aplikasi *game* papan berbasis *augmented reality*. Lawan pada *Game* ini akan diberikan *Artificial Intelligence* agar dapat bermain melawan pemain. Akan dilakukan tahap desain *game*, *markerless*, dan lingkungan yang dibutuhkan.

c. Tahap III

Model yang telah diproses pada tahap sebelumnya akan diterjemahkan ke desain aplikasi yang akan di *coding* dengan Javascript dan C#. Setelah proses tersebut selesai dibuat, maka akan dimulai untuk masuk ke *coding*.

d. Tahap IV

Pada tahap ini, proses *coding* C# dilakukan di Unity3D. Hasil dari proses ini adalah sebuah aplikasi AR yang telah terbuat. Pada proses ini masuk ke *constructions* saat penggunaan model RAD.

e. Tahap V

Tahap ini meliputi pengujian aplikasi. Pengujian dijalankan ketika pemain mencoba permainan dengan menginstall aplikasi ke *smartphone*. Selanjutnya, pemain membuka aplikasi dan mengarahkan kamera *smartphone* ke *Markerless* yang telah dibuat. Akhirnya, pemain dapat memainkan permainan tersebut.

3.2. Model Pengembangan Perangkat Lunak

RAD meliputi tiga hal, yaitu:

a. *Requirements Planning*

Pada tahap ini, analis akan melakukan proses menemukan kebutuhan yang diperlukan untuk membuat aplikasinya nanti serta melakukan analisis hasil dari *game* dan mengidentifikasi data yang dibutuhkan.

b. *User Design*

Pada tahap ini dilakukan proses dari desain aplikasi sesuai dengan data yang diberikan oleh analis. Hasil dari proses ini adalah berupa *layout* aplikasi *game* Othello yang akan dibangun, mulai dari ukuran papan, warna papan, peletakan teks dan warna dari bidak yang

akan dimainkan. Proses desain ini juga akan melibatkan *user* yang terlibat pada proses sebelumnya, apakah desain yang diberikan sudah sesuai kesepakatan.

c. Konstruksi

Bahasa pemrograman yang akan dipakai setelah tahap desain dilakukan menggunakan C# dengan tools *Game Engine* Unity3D dan Vuforia.

d. Testing / Evaluasi

Bagian ini dilakukan dengan mencoba aplikasi yang di *coding* ke pengguna pada proses konstruksi.

4. Hasil dan Diskusi

4.1. *GamePlay* dan Algoritma Permainan

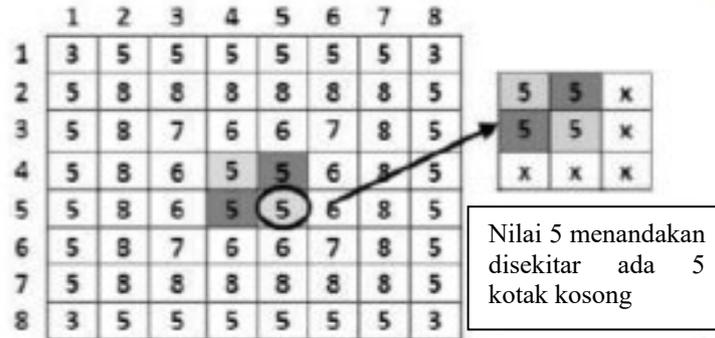
Game Othello menggunakan bidak putih dan hitam di atas *board game* berbentuk yang memiliki ukuran 8x8. Cara bermain Othello sangatlah mudah, dengan jumlah empat bidak saat awal permainan (2 bidak putih dan 2 bidak hitam), pemain harus mengumpulkan bidak warna yang dimainkan hingga seluruh *board game* berhasil terisi oleh bidak pemain. Langkah pertama permainan dapat menentukan kemungkinan pergerakan lawan sehingga dapat digunakan untuk mengendalikan permainan. Proses ini dipakai agar mendeteksi semua bidak di atas *smart board* sampai data tersebut diproses melakukan pencarian pergerakan yang akan digunakan dan menjadi inputan proses di algoritma Alpha Beta Pruning [16]. Beberapa bagian yang ditambahkan didalam artikel ini adalah setiap kemungkinan pergerakan yang dihasilkan dari Minimax dan Alpha Beta Pruning, akan langsung ditampilkan ke pemain dan lawan. Setelah pemain menentukan pilihan, nilai *edge table* akan langsung di-*generate* agar beberapa pilihan langkah selanjutnya akan langsung ditampilkan setelah lawan membuat pergerakan. Visual dan tampilnya informasi posisi pergerakan bidak yang boleh dilewati didapatkan dari hasil dari nilai *edge*, *liberties* dan *corner* yang kemungkinan penelusuran cabangnya sudah dibatasi dengan Alpha Beta Pruning. Gambar 3 menunjukkan contoh inisialisasi permainan.

I\J	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	2	1	0	0	0
4	0	0	0	1	2	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Gambar 3. Inisialisasi Permainan

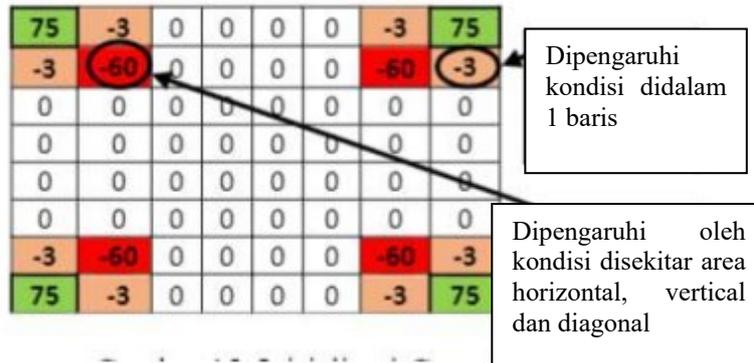
Pengecekan pergerakan data pada papan akan membantu pemain berbasis AI (Lawan) membuat keputusan pilihan dimana bidaknya akan diletakan. Selain itu lawan dapat menghindari penempatan bidak di papan secara sembarangan. Arah pergerakan tetangga diperoleh dari analisis posisi tetangga seperti gambar dibawah. Gambar 4 menunjukkan contoh pengecekan posisi pergerakan.

yang sangat buruk akan semakin kecil. Semakin besar nilai *mobility*, itu juga berarti semakin banyak kebebasan menentukan gerakan yang baik. Model ini nantinya akan digunakan untuk menangani kemungkinandari proses *mobility*. Gambar 7 menunjukkan contoh perhitungan *Liberties*.



Gambar 7. Perhitungan *Liberties*

Fokus terhadap *Corner* di sini juga memiliki peranan yang berpengaruh cukup tinggi. Cara untuk memproses *Corner* adalah dengan meletakkan nilai tinggi pada bagian *Corner* dari pada bagian lain yang berbatasan. Gambar 8 menunjukkan contoh penguasaan sudut papan permainan.



Gambar 8. Penguasaan bidak di posisi sudut

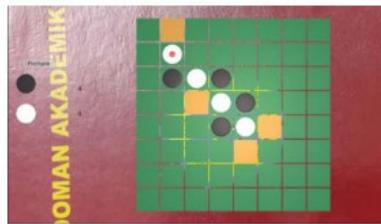
4.2. Hasil Tampilan Aplikasi

Penentuan jarak antara target dan kamera akan berpengaruh terhadap kecepatan tampilnya objek *Augmented reality* yang diinginkan. Semakin optimal jarak kamera ke target, semakin baik pula kinerja aplikasinya. Target gambar (Penanda) merupakan logo dari universitas yang terdapat pada Gambar 9.



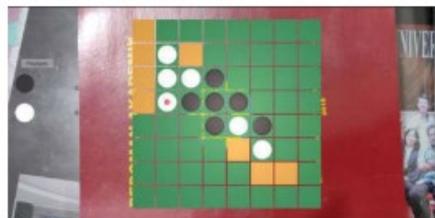
Gambar 9. Target Gambar / Penanda

Gambar 10 adalah hasil yang didapat saat pemain membuka aplikasi dan mengarahkan kameranya keatas target gambar dan dapat memulai permainannya melawan AI yang telah diberikan algoritma Minimax.



Gambar 10. Tampilan Aplikasi via Smartphone

Gambar 11 adalah hasil yang didapat dari hasil nilai *edge* dan *mobility* yang berwarna kuning. Warna kuning tersebut menandakan posisi papan yang dapat diisi oleh bidak player dan AI. Sehingga pemain baru pun dapat mengerti akan bergerak ke posisi mana (warna kuning).



Gambar 11. Tampilan Hasil Nilai Edge dan Mobility

Gambar 12 adalah hasil yang didapat dari hasil nilai *liberties*, Penguasaan *Corner* di sini juga sangat berpengaruh, karena bidak diposisi tersebut tidak dapat diubah atau diputar. Pemain dapat memilih posisi ini jika dianggap pemain adalah posisi terbaik. Sedangkan AI dapat secara otomatis memilih posisi sesuai algoritma Minimax yang dihasilkan.



Gambar 12. Tampilan Hasil Penguasaan Sudut

Gambar 13 adalah hasil yang didapat dari hasil permainan yang telah diselesaikan. Pemain dianggap menang jika bidak putih berjumlah lebih banyak dari bidak hitam, begitu juga sebaliknya untuk kekalahan pemain.



Gambar 13. Tampilan Menang (Kiri) dan Kalah (Kanan)

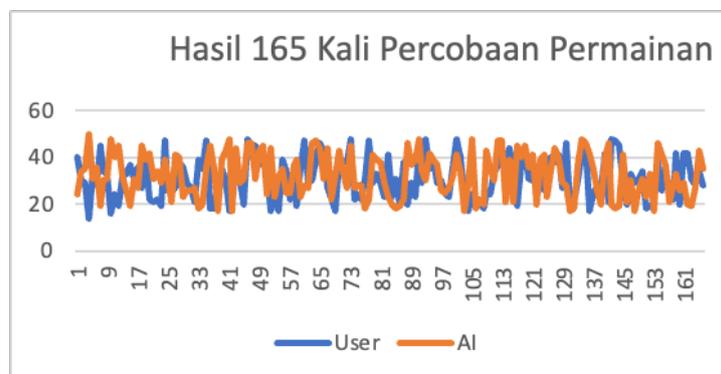
4.3. Pengujian Aplikasi

Pengujian dilakukan sebanyak 165 kali dengan 10 pengguna yang menggunakan *smartphone* dengan jarak antara *smartphone* dan target gambar 20-40 cm, sehingga *game* dapat

di-load oleh *marker* dengan baik. Dari percobaan tersebut beberapa pemain mengalami kemenangan dan kekalahan, yang terlihat dibawah ini.

Tabel 1. Perbandingan Permainan Pemain dan Non Player Character

No.	Pemain	Jumlah Bidak Pemain	Jumlah Bidak Lawan (Minimax-AI)
1	1	24	40
2	1	35	29
3	2	30	34
4	3	40	24
5	4	28	36
6	4	48	16
7	5	45	20
8	6	50	14
9	6	34	30
10	6	43	21
11	7	20	44
12	8	36	28
13	9	23	41
14	10	14	50
15	4	53	11
...	
164	7	37	27
165	8	31	33



Gambar 14. Hasil percobaan dimana AI berhasil Menang dengan presentase 63% dari *user*

5. Kesimpulan dan Saran

Kesimpulan yang tercapai setelah melewati proses RAD yaitu dengan menggunakan algoritma Minimax-Alpha Beta Pruning berbasis *Augmented reality* adalah permainan yang awalnya tradisional akhirnya dapat dibuat berbasis teknologi masa kini, sehingga permainannya menjadi lebih menarik. Algoritma Minimax yang dikembangkan bersama *Alpha Beta Pruning* mampu membuat lawan juga mampu berfikir layaknya manusia sehingga kecerdasan buatan didalamnya cocok untuk diterapkan. Hasil 63% yang dicapai dalam percobaan dalam keberhasilan mengalahkan pemain manusianya. *Augmented reality* yang digunakan membuat suasana bermain *game* menjadi lebih interaktif dengan menggunakan fasilitas kamera pada *smartphone*. Saran untuk aplikasi ini yaitu dapat lebih dikembangkan lagi seperti informasi terkait penyimpanan skor dan peningkatan kemampuan kecerdasan NPC (*Non Player Character*) dalam melawan pemain manusia.

Referensi

- [1] Machfuz. (2017). *Aplikasi Peta Interaktif Berbasis Teknologi Augmented reality Kawasan Pariwisata Pulau Bawean*. Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM), ITS
- [2] Calvinus W., (2017). *Implementasi Algoritma Minimax Dalam Game Othello*. Jurnal Teknik Informatika Unika St. Thomas (JTIUST), Volume 02 Nomor 01, Juni 2017.
- [3] Indra, E., (2020). *Analisa Efektivitas Algoritma Minimax, Alpha Beta Pruning, dan Negamax dalam Penerapannya pada Permainan Papan (Board Game)*. Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI) Vol. 3 No. 2 (2020): Jurnal Ilmu Komputer dan Sistem Informasi
- [4] Brian Sumali. (2016). *Implementation of Minimax with Alpha-Beta Pruning as Computer Player in Congklak*. Jurnal Teknik Informatika dan Sistem Informasi Volume 2 Nomor 2 Agustus 2016.
- [5] Yang, (2019). *Expectation Minimax Algorithm for the Two-Player Military Chess Game*. Chinese Control And Decision Conference (CCDC) IEEE
- [6] Jesse Schell. (2019). *The Art of Game Design: A Book of Lenses, Third Edition*. A K Peters / CRC Press
- [7] Santiago Videgaína, (2021), *Performance Study of Minimax and Reinforcement Learning Agents Playing the Turn-based Game Iwoki*. Applied Artificial Intelligence: An International Journal
- [8] Minsong Liu. (2020). *An Improved Minimax-Q Algorithm Based on Generalized Policy Iteration to Solve a Chaser-Invader Game*. Auckland University of Technology. IEEE Xplore.
- [9] Kurniawan, A. Pamungkas. (2016), *Algoritma Minimax Sebagai Pengambil Keputusan dalam Game Tic-Tac-Toe*. Semin. Nas. Teknol. Inf. dan Multimed.
- [10] Paul M., (2018), *Virtual & Augmented reality For Dummies (For Dummies (Computer/Tech))*. For Dummies; 1st edition.
- [11] Jonathan L., (2017) , *Augmented reality for Developers: Build practical augmented reality applications with Unity, ARCore, ARKit, and Vuforia*. Packt Publishing
- [12] Matthias F., (2018), *How to Design Programs, second edition: An Introduction to Programming and Computing*, London : The MIT Press.
- [13] Paul L. *RGB-D Image Analysis and Processing (Advances in Computer Vision and Pattern Recognition)*. Springer; 1st ed. 2019 edition
- [14] Fernando, Mario, (2013), *Membuat Aplikasi Android Augmented reality Menggunakan Vuforia SDK dan Unity*. Solo : Buku AR Online.
- [15] Graham Harman. 2018. *Object-Oriented Ontology: A New Theory of Everything*. Pelican; Illustrated edition
- [16] Jeon (2018). *An Implementation of Othello Game Player Using ANN based Records Learning and Minimax Search Algorithm*. The Transactions of The Korean Institute of Electrical Engineers