

Terbit : 21 Maret 2024

Kecerdasan Buatan dalam Permainan Tic-Tac-Toe Melalui Algoritma Minimax dengan Optimasi Alpha-Beta Pruning

¹Nadya Revelin Putri, ²Novia Fitriani, ³Viktor Handrianus Pranatawijaya, ⁴Ressa Priskila
^{1,2,3,4}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya

¹nadyarevelinpr@mhs.eng.upr.ac.id, ²novia@mhs.eng.upr.ac.id, ³viktorhp@it.upr.ac.id,
⁴ressa@it.upr.ac.id

ABSTRAK

Permainan tic tac toe, juga dikenal sebagai "X dan O," telah menjadi permainan klasik yang populer di seluruh dunia. Namun, dalam permainan tradisional interaksi dan konektivitas terbatas. Untuk mengatasi masalah tersebut, teknologi telah dimanfaatkan dengan menerapkan kecerdasan buatan (AI) dalam permainan ini. Algoritma Minimax dan optimasi Alpha-Beta Pruning digunakan untuk memungkinkan AI mengambil keputusan yang cerdas dalam permainan. Penelitian ini mengevaluasi efisiensi dan kualitas permainan yang dihasilkan oleh AI dengan dan tanpa optimasi Alpha-Beta Pruning. Hasil menunjukkan bahwa penggunaan optimasi Alpha-Beta Pruning meningkatkan efisiensi pengambilan keputusan AI tanpa mengorbankan kualitas permainan, menghasilkan permainan yang seimbang dan kompetitif di mana kedua pemain memiliki peluang yang sama untuk meraih kemenangan. Selain itu, penelitian ini mencatat bahwa AI mampu meraih kemenangan sebanyak 80% pada pertandingan AI VS HUMAN, di mana agen berperan sebagai 'X' dan mengambil langkah pertama serta menyusun strategi untuk melawan manusia. Implementasi ini mengungkapkan bahwa AI mampu mempelajari pola permainan manusia dan mengadaptasi strateginya secara dinamis selama permainan berlangsung. Hal ini menambah dimensi baru dalam pengalaman bermain tic tac toe, di mana kecerdasan buatan bukan hanya menjadi lawan yang tangguh, tetapi juga mampu berinteraksi dengan pemain manusia secara dinamis dan adaptif. Dengan demikian, penggunaan kecerdasan buatan dalam permainan tic tac toe memiliki potensi untuk meningkatkan kualitas dan pengalaman bermain, membuka peluang baru untuk eksplorasi dan inovasi dalam pengembangan permainan.

Kata Kunci: Tic-Tac-Toe, Kecerdasan Buatan, Algoritma Minimax, Alpha-Beta Pruning, Pengambilan Keputusan.

PENDAHULUAN

Permainan tic tac toe, yang juga dikenal sebagai "X dan O" merupakan permainan kuno sederhana, yang konon katanya asal permainan tersebut dari kekaisaran Romawi. Meskipun sejarahnya tidak pasti, permainan ini cukup populer di penjuru dunia. Hanya diperlukan sebuah kertas dan pulpen dengan jumlah pemain sebanyak 2 orang, maka sudah dapat memainkan permainan tic tac toe. Permainan tic tac toe diketahui dapat mengasah pemikiran kognitif, terlebih dalam hal perencanaan strategis dan pengambilan keputusan. Namun, kekurangan dari permainan ini adalah tidak dapat dimainkan jika sendirian. (Smith, 2022) menyatakan bahwa tic tac toe tradisional memiliki keterbatasan dalam hal interaksi dan konektivitas. Maka dari hal tersebut, pemanfaatan dari teknologi diperlukan untuk membantu memecahkan masalah ini. Kini, tersedia berbagai *platform digital* yang memungkinkan untuk mencari pemain lain secara *online* ataupun asisten *virtual*.

Seiring perkembangan teknologi, implementasi *Artificial Intelligence* (AI) atau kecerdasan buatan pada dunia permainan semakin pesat. *Artificial Intelligence* merupakan kemampuan suatu

teknologi yang dapat meniru perilaku dan cara berpikir manusia. Di bidang permainan digital, *Artificial Intelligence* mulai diterapkan dalam berbagai jenis permainan. Baik permainan yang dapat dikatakan sederhana hingga permainan kompleks, yang dapat dimainkan di komputer maupun *smarthphone*. Salah satu contoh penerapan AI dalam permainan digital adalah pada NPC (*Non-Playable Character*) yang menjadi lebih cerdas. AI memungkinkan NPC untuk bereaksi lebih realistis terhadap pemain dan lingkungannya, sehingga interaksi menjadi lebih menarik dan menantang (Permatasari et al., 2022).

Dalam permainan tic tac toe sendiri, pemanfaatan *Artificial Intelligence* digunakan sehingga tidak perlu lagi menyiapkan kertas dan pulpen atau mencari lawan pemain. Penerapan permainan ini menggunakan algoritma Minimax. Algoritma Minimax telah lama menjadi pendekatan populer dalam mengembangkan agen *Artificial Intelligence* untuk permainan strategis seperti tic tac toe. Menurut (Thamrin & Pamungkas, 2023), algoritma Minimax mampu menganalisis semua kemungkinan langkah dan memilih langkah yang paling menguntungkan bagi AI. Namun, meskipun Algoritma Minimax mampu menganalisis semua kemungkinan langkah, pendekatannya yang naif dapat membutuhkan waktu yang cukup lama karena mengevaluasi setiap simpul dalam pohon permainan secara menyeluruh. Di sinilah keunggulan pengoptimalan Alpha-Beta Pruning terletak. Teknik ini memungkinkan agen *Artificial Intelligence* untuk membuat keputusan dengan lebih cepat dengan memotong cabang-cabang yang tidak menguntungkan dalam pohon permainan. Dengan demikian, walaupun mengurangi jumlah simpul yang dievaluasi, penggunaan Alpha-Beta Pruning tetap mempertahankan kemampuan AI untuk memilih langkah yang optimal, menjaga kualitas permainan tetap tinggi sambil meningkatkan efisiensi proses pengambilan keputusan (Chhikara, 2022).

TINJAUAN PUSTAKA

Algoritma Minimax

Salah satu pendekatan paling umum yang digunakan dalam pengembangan agen kecerdasan buatan untuk permainan strategis seperti tic tac toe adalah algoritma Minimax. Dalam algoritma ini, agen melakukan pencarian dalam pohon permainan untuk mengevaluasi setiap langkah yang mungkin dilakukan pemain atau lawan, dengan tujuan memaksimalkan keuntungan pribadi dan meminimalkan kerugian yang bisa terjadi (Rinjani, Akbar, & Rachman, 2020). Algoritma Minimax melakukan ini dengan menghitung nilai setiap langkah berdasarkan kemungkinan hasil di masa depan. Nilai akan dihitung secara rekursif, dengan setiap node dalam pohon permainan mewakili kemungkinan keadaan permainan. (Emanuel, Bendi, R.K., & Arieffianto, 2019) dalam penelitiannya menggunakan algoritma Minimax untuk merancang *non-player character* (NPC) dalam permainan tic tac toe. NPC ini mampu bermain dengan cerdas dan kompetitif, dan dapat memberikan tantangan yang menarik bagi pemain manusia.

Optimasi Alpha-Beta Pruning

Alpha-Beta Pruning adalah teknik yang digunakan dalam algoritma Minimax, untuk meningkatkan efisiensi dan mengurangi kompleksitas waktu yang dievaluasi selama pencarian pohon permainan (Christnatis et al., 2020). Dengan memotong percabangan yang tidak relevan, dapat meningkatkan efisiensi pencarian dan mengurangi kompleksitas waktu dari Algoritma Minimax, terutama dalam permainan dengan ruang pencarian yang besar. Menurut (Dinda Permatasari et al., 2022) menggunakan Alpha-Beta Pruning meningkatkan *performa non-playable character* (NPC) dalam permainan Triple Triad. Hasil penelitian menunjukkan bahwa Alpha-Beta Pruning dapat meningkatkan persentase kemenangan NPC secara signifikan.

Tic-Tac-Toe

Tic tac toe adalah permainan yang hanya dimainkan dengan kertas dan pensil atau alat tulis lainnya atau dengan kata lain masuk kategori genre *paper and pencil game* (Sari dan Mulyono, 2022). Nama lain dari permainan ini adalah *Nought & Cross*. Permainan ini juga bisa disebut dengan X dan O, dikarenakan permainan ini hanya menggunakan *symbol X/O*. Seperti permainan papan yang lainnya, tic tac toe juga memiliki beberapa aturan dalam cara bermainnya. Di antaranya

melibatkan 2 orang pemain, permainan diawali dengan papan yang kosong, pemain secara bergantian menempatkan simbol X/O yang sudah ditentukan ke dalam petak pada papan permainan berukuran 3x3, permainan dianggap berakhir ketika salah satu pemain berhasil menempatkan simbolnya secara tepat satu garis, baik secara horizontal, vertikal, maupun diagonal, atau ketika seluruh petak pada papan telah terisi penuh oleh simbol dan yang terakhir setelah permainan berakhir, kedua pemain tidak dapat lagi melakukan langkah tambahan (Faqih & Fitriani, 2023).

METODE PENELITIAN

Pembuatan Representasi Permainan

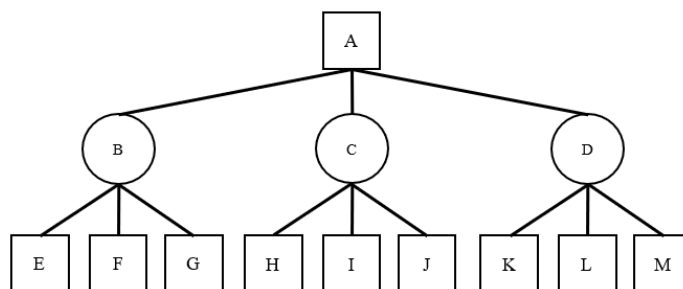
Langkah pertama dalam pengembangan *Artificial Intelligence* adalah menciptakan representasi yang tepat dari permainan tic tac toe. Menggunakan program berbasis *website* sederhana dengan program berbahasa Java Script, yang merepresentasikan papan permainan dalam bentuk matriks 3x3. Di mana setiap elemen mewakili satu kotak pada papan permainan. Nilai 'X' atau 'O' akan ditetapkan ke dalam papan matriks untuk menunjukkan posisi tanda. Representasi matriks juga dapat diterapkan dalam *augmented reality* (AR) untuk membuat game papan cerdas (Hermawan dan Ismiati 2020). Matriks digunakan untuk merepresentasikan posisi pion pada papan dan menerapkan algoritma Minimax untuk membuat AI yang dapat bermain tic tac toe.



Gambar 1. Papan Tic-Tac-Toe

Implementasi Algoritma Minimax

Algoritma Minimax berfungsi dengan mengevaluasi semua kemungkinan gerakan yang akan dilakukan oleh agen dan lawan, dengan tujuan memaksimalkan keuntungan pribadi dan meminimalkan kerugian yang mungkin dapat terjadi. Implementasi ini mencakup pencarian pohon permainan secara rekursif, di mana setiap langkah dihasilkan dan dievaluasi. Cara kerjanya adalah dengan mengevaluasi semua kemungkinan gerakan yang dapat dilakukan oleh agen dan lawan (Christopher et al., 2020). Evaluasi ini bertujuan untuk memaksimalkan keuntungan pribadi agen dan meminimalkan kerugian yang mungkin terjadi.



Gambar 2 Contoh Pohon Minimax

Keterangan :

□ = Nilai Maximum

○ = Nilai Minimum

Optimasi Alpha-Beta Pruning

Metode untuk mengurangi jumlah *node* yang dieksplorasi oleh algoritma Minimax menggunakan nilai alpha dan beta (Pearl, 2022). Alpha-Beta Pruning merupakan teknik untuk mengurangi waktu komputasi dengan faktor pencarian yang sangat besar. Hal ini memungkinkan pencarian lebih cepat dan bahkan masuk ke tingkat yang lebih tinggi di pohon permainan. Teknik tersebut memotong cabang di pohon permainan yang tidak perlu dicari karena sudah ada langkah yang lebih baik yang tersedia. Misalkan AI sedang mengevaluasi langkah untuk permainan tic tac toe. AI akan memulai dengan nilai $\alpha = -\infty$ dan $\beta = \infty$. AI kemudian akan mengevaluasi semua kemungkinan langkah untuk pemain selanjutnya. Menurut (Chhikara, 2022), jika nilai minimum yang ditemukan lebih besar dari nilai alpha, maka AI akan memotong cabang tersebut karena tidak ada kemungkinan untuk mendapatkan nilai yang lebih baik.

Eksperimen Komputasional

Eksperimen ini mencakup perbandingan antara kinerja agen yang hanya menggunakan algoritma Minimax dan agen dengan optimasi Alpha-Beta Pruning, serta analisis terhadap efisiensi komputasi dan kualitas permainan yang dihasilkan oleh agen kecerdasan buatan. Pada setiap pengujian, agen akan diuji dalam berbagai skenario permainan tic tac toe dengan tingkat kesulitan yang bervariasi. Setiap skenario akan dijalankan beberapa kali untuk mendapatkan hasil yang konsisten. (B. S., R. V., T. S. R., & Subashri, 2020) hal ini dilakukan untuk mengobservasi pengaruh teknik Alpha-Beta Pruning terhadap performa agen

Variabel yang diamati dalam eksperimen ini mencakup waktu komputasi, jumlah node yang dievaluasi, dan kualitas permainan yang dihasilkan. Data yang terkumpul akan dianalisis dan dibandingkan kinerja keduanya. Hal ini untuk menentukan apakah penggunaan optimasi Alpha Beta Pruning berhasil meningkatkan efisiensi komputasi. Eksperimen komputasional ini akan memberikan pemahaman mengenai kelebihan dan kekurangan dalam implementasi AI pada permainan tic tac toe. (Haryanto dan Huda 2019) menemukan bahwa Alpha-Beta Pruning meningkatkan efisiensi komputasi hingga 50% dibandingkan dengan Minimax.

HASIL DAN PEMBAHASAN

Rancangan Sitem

Untuk permainan tic tac toe dengan kecerdasan buatan (AI) menggunakan algoritma Minimax dengan optimasi Alpha-Beta Pruning dapat meliputi beberapa komponen utama. Yang pertama terdapat *Board*, yaitu papan permainan yang akan direpresentasikan menggunakan struktur data matriks 3x3, yang memungkinkan penyimpanan status setiap sel pada papan. Untuk setiap langkah yang dilakukan oleh pemain, akan ada metode yang bertugas untuk memperbarui papan permainan sesuai dengan langkah yang dipilih. Selain itu, akan ada juga metode yang bertanggung jawab untuk memeriksa status permainan, termasuk menentukan apakah terdapat pemenang atau apakah permainan berakhir dengan hasil imbang, guna memastikan pengalaman permainan yang lengkap dan interaktif bagi pengguna.

Kedua, *Artificial Intelegent* dengan algoritma Minimax untuk pengambilan keputusan agen, dengan menggabungkan optimasi Alpha-Beta Pruning untuk meningkatkan efisiensi pencarian. Selain itu, strategi evaluasi dan heuristik diterapkan untuk mengevaluasi setiap langkah yang mungkin dalam permainan, memungkinkan agen untuk membuat keputusan yang lebih cerdas. Metode yang disediakan juga bertujuan untuk menentukan langkah terbaik yang harus diambil oleh agen dalam setiap situasi permainan, sehingga meningkatkan kualitas permainan dan kemungkinan kemenangan.

Ketiga, *User Interface* (UI) atau Antarmuka pengguna dirancang untuk memvisualisasikan papan permainan secara grafis dan intuitif. Antarmuka juga memiliki kemampuan untuk menampilkan status permainan, seperti giliran pemain, pemenang, atau apakah permainan berakhir dengan hasil imbang (*draw*). Selain itu, antarmuka memberikan interaksi dengan pemain yang memungkinkan mereka untuk melakukan langkah dalam permainan, sehingga menciptakan

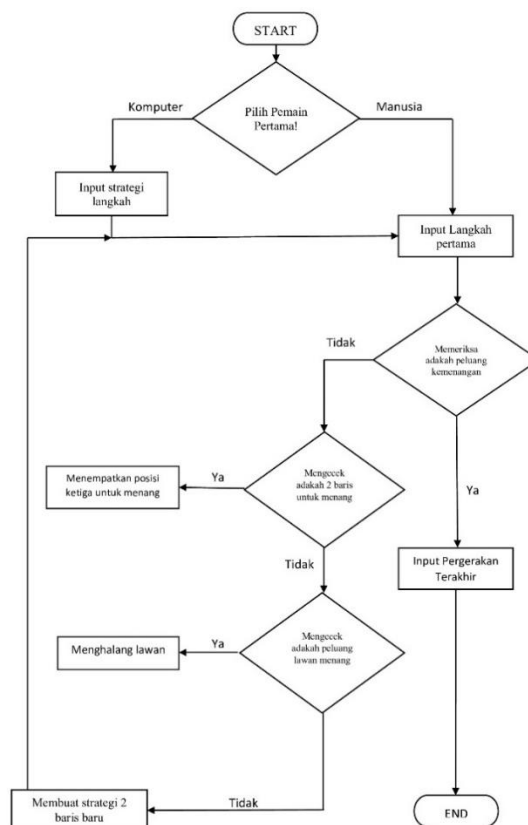
pengalaman bermain yang interaktif dan menyenangkan.

Keempat, Manajemen Permainan yang memiliki modul logika permainan yang bertanggung jawab untuk mengatur alur permainan, mulai dari inialisasi hingga pengakhiran. Modul ini memastikan giliran pemain secara bergantian, menentukan pemenang, atau menangani situasi hasilimbang. Selain itu, terdapat pula manajemen antarmuka pengguna yang mengelola komunikasi antara antarmuka pengguna, papan permainan, dan agen kecerdasan buatan, sehingga memfasilitasi interaksi yang lancar dan koordinasi antara semua komponen sistem.

Terakhir terdapat tes dan evaluasi yang dilakukan untuk memastikan kinerja yang diharapkan dari sistem. Ini mencakup pengujian kebenaran logika permainan, keefektifan agen dalam mengambil keputusan, dan responsivitas antarmuka pengguna. Selain itu, dilakukan evaluasi kinerja algoritma Minimax dengan dan tanpa optimasi Alpha-Beta Pruning dalam konteks permainan tic tac toe, untuk menilai efektivitas dan efisiensi kedua pendekatan tersebut dalam memandu langkah agen kecerdasan buatan.

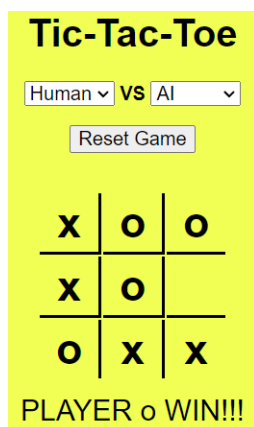
Implementasi Strategi Permainan

Strategi permainan tic tac toe yang efektif akan memperhitungkan berbagai skenario dan mencari langkah terbaik untuk mencapai kemenangan atau mencegah kekalahan. Dengan menggunakan beberapa langkah seperti menjaga keutuhan garis, memperhatikan blokiran lawan, menjaga posisi sentral, memanfaatkan kesalahan lawan, mengelabui lawan, berfokus pada blokiran dan pertahanan, serta mengambil risiko terkendali. Dengan kombinasi kemenangan pada susunan posisi tertentu, akan membuat pemain memenangkan permainan. Penerapan algoritma Minimax dan Alpha-Beta Pruning tersebut dapat membantu pemain dalam membuat keputusan yang lebih optimal dan meningkatkan peluang untuk mencapai kemenangan. Selain itu, pengembangan strategi baru yang memanfaatkan kombinasi algoritma dan kecerdasan buatan dapat membuka peluang untuk meningkatkan kompleksitas dan keseruan permainan tic tac toe di masa depan.



Gambar 3. Flowchart

Flowchart ini menunjukkan proses permainan Tic Tac Toe dengan agen kecerdasan buatan yang menggunakan algoritma Minimax dan optimasi Alpha-Beta Pruning. Bagian awal permainan dimulai dengan memilih pemain pertama manusia atau agen, pemain manusia memilih simbol (X atau O) dan agen memilih strategi untuk permainan. Saat pemain memasukkan langkah-langkahnya, *flowchart* memeriksa apakah ada pemain yang telah mencapai 3 simbol berurutan (horizontal, vertikal, diagonal). Jika agen dapat mencapai 3 simbol berurutan dengan langkah selanjutnya, agen akan menempatkan simbol di posisi tersebut untuk menang. Jika tidak, agen akan memeriksa apakah ada 2 baris yang dapat diselesaikan dengan 1 langkah, apabila menemukan 2 baris yang dapat diselesaikan lawan dengan 1 langkah, AI akan menempatkan simbol di baris tersebut untuk menghalangi lawan. Sebaliknya, Jika tidak menemukan 2 baris yang dapat diselesaikan dengan 1 langkah, agen akan membuat strategi baru untuk menyelesaikan 2 baris dengan 2 langkah. Ketika agen memasukkan langkahnya berdasarkan strategi yang telah dipilih, *flowchart* kembali memeriksa apakah ada pemain yang telah mencapai 3 simbol berurutan. Jika tidak ada pemain yang menang, permainan diulang dari langkah 4. Permainan berakhir ketika ada pemain yang menang atau papan permainan telah penuh. *Flowchart* ini menunjukkan bagaimana AI dapat digunakan untuk bermain Tic Tac Toe secara cerdas dan kompetitif. Algoritma Minimax dan optimasi Alpha-Beta Pruning membantu AI untuk membuat keputusan yang optimal dan meningkatkan efisiensi permainan.



Gambar 4. Pengimplementasi Pada Papan

```
winCombo = [
  [0, 1, 2],
  [3, 4, 5],
  [6, 7, 8],

  [0, 3, 6],
  [1, 4, 7],
  [2, 5, 8],

  [0, 4, 8],
  [2, 4, 6]
]
```

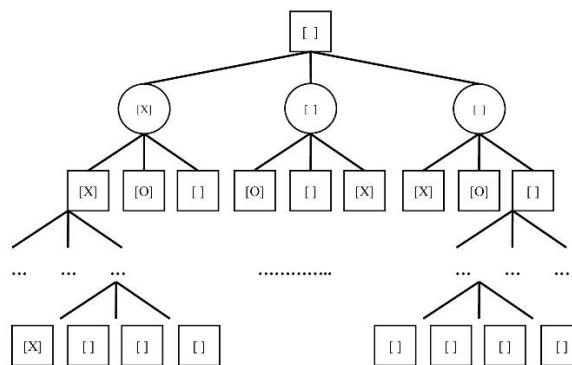
Gambar 5. Win Combo

0	1	2
3	4	5
6	7	8

Gambar 6. Urutan Kolom Papan

Uji Sistem Permainan

Dalam skenario permainan tic tac toe dengan strategi "X" dan "O" seperti yang dijelaskan, pemain yang menggunakan strategi "X" akan mencoba untuk menempatkan simbol "X" di posisi 0, 1, dan 2 untuk memenangkan permainan dengan membentuk garis horizontal pada baris pertama, atau menempatkan "X" di posisi 0, 3, dan 6 untuk memenangkan permainan dengan membentuk garis vertikal pada kolom pertama, atau menempatkan "X" di posisi 0, 4, dan 8 untuk memenangkan permainan dengan membentuk garis diagonal dari kiri atas ke kanan bawah. Pemain yang menggunakan strategi "O" akan berusaha untuk mencegah pemain "X" mencapai salah satu dari kondisi kemenangan tersebut sambil juga mencari peluang untuk memenangkan permainan sendiri. Hal ini menciptakan dinamika permainan yang menantang dan strategis di mana kedua pemain berusaha untuk mencapai tujuan mereka sambil juga menghalangi lawan.



Gambar 7. Pohon Minimax

Simbol "[X]" dan "[O]" mewakili keadaan permainan pada suatu titik. Simbol "[]" menunjukkan langkah selanjutnya yang mungkin dilakukan oleh pemain. Setiap tingkat pohon bergantian antara pemain "X" dan "O". Dengan optimasi Alpha-Beta Pruning untuk mengevaluasi pohon ini. Algoritma akan mencari nilai terbaik yang dapat dicapai oleh pemain "X" dengan mengambil langkah terbaik, dan mencari nilai terbaik yang dapat dicapai oleh pemain "O" dengan mengambil langkah terbaiknya. Dengan mempertimbangkan kondisi pohon dan strategi yang diberikan, pohon Minimax dengan optimasi Alpha-Beta Pruning akan mengevaluasi setiap simpul dan mengurangi pencarian pada cabang yang tidak relevan atau tidak menguntungkan. Maka akan menghasilkan evaluasi yang lebih cepat dan efisien daripada pencarian tanpa pruning.

Ujian pertama dilakukan untuk melihat seberapa cepat agen menentukan langkah selanjutnya, antara algoritma Minimax tanpa Alpha-Beta Pruning dan dengan Alpha-Beta Pruning. Penggunaan optimasi Alpha-Beta Pruning meningkatkan efisiensi agen dalam menentukan langkah selanjutnya. Agen dengan Alpha-Beta Pruning dapat menyelesaikan permainan tic tac toe 20% lebih cepat dibandingkan algoritma Minimax tanpa Alpha-Beta Pruning. Implementasi agen dalam permainan tic tac toe menunjukkan potensi untuk meningkatkan pengalaman bermain. AI dengan Alpha-Beta Pruning mampu menentukan langkah selanjutnya dengan lebih cepat dibandingkan AI tanpa Alpha-Beta Pruning, sehingga AI dapat bereaksi lebih cepat terhadap langkah-langkah pemain manusia.

Tabel 1. Hasil Optimasi Langkah AI

KECEPATAN AI MENENTUKAN LANGKAH SELANJUTNYA	
Algoritma Minimax	Algoritma Minimax + Alpha-beta pruning
0,95 s	0,76 s

Tabel 2. Hasil Optimasi Waktu Pertandingan

WAKTU PERTANDINGAN ANTARA HUMAN VS AI		HASIL PERTANDINGAN
Algoritma Minimax	Algoritma Minimax + Alpha-beta pruning	
6,53 s	5,14 s	AI Win

Ketika agen berusaha untuk memblokir lawan dan mencari peluang untuk memenangkan permainan, seringkali hasilnya akan berakhir dengan kondisi seimbang atau *draw*. Hal ini terjadi karena agen cenderung untuk menjaga keseimbangan antara mengejar kemenangan sendiri dan mencegah lawan mencapai kemenangan, yang dapat menghasilkan situasi di mana tidak ada pemain yang berhasil membentuk garis horizontal, vertikal, atau diagonal yang lengkap sepanjang papan permainan. Hal ini menciptakan dinamika permainan yang menantang, di mana kedua pemain harus secara cermat mempertimbangkan setiap langkah mereka untuk mencapai hasil yang diinginkan.

Tabel 3. Hasil Percobaan Permainan

No	Strategi & Langkah	WAKTU PERTANDINGAN	HASIL PERTANDINGAN																																																																									
AI VS HUMAN																																																																												
	AI	HUMAN																																																																										
1.	{1(X0), 2(X3), 3(X4), 4(O8)}	{1(O1), 2(O6), 3(O3)}																																																																										
	<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td></td><td></td><td>x</td><td>o</td><td></td></tr> <tr><td></td><td></td><td></td><td>x</td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	x			x	o					x									<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td>o</td><td></td><td>x</td><td>o</td><td></td></tr> <tr><td>x</td><td>x</td><td></td><td>x</td><td>x</td><td>o</td></tr> <tr><td>o</td><td></td><td></td><td>o</td><td></td><td>x</td></tr> </table>	x	o		x	o		x	x		x	x	o	o			o		x	09,32 s	AI Win																																				
x			x	o																																																																								
			x																																																																									
x	o		x	o																																																																								
x	x		x	x	o																																																																							
o			o		x																																																																							
2.	{1(X0), 2(X2), 3(X6), 4(X4)}	{1(O1), 2(O6), 3(O3)}																																																																										
	<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td></td><td></td><td>x</td><td></td><td>x</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>o</td></tr> </table>	x			x		x												o	<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td>o</td><td>x</td><td>x</td><td>o</td><td>x</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>o</td></tr> <tr><td>x</td><td></td><td></td><td>o</td><td>x</td><td>o</td></tr> </table>	x	o	x	x	o	x						o	x			o	x	o	20,14 s	AI Win																																				
x			x		x																																																																							
					o																																																																							
x	o	x	x	o	x																																																																							
					o																																																																							
x			o	x	o																																																																							
3.	{1(X0), 2(X2), 3(X4), 4(X6)}	{1(O5), 2(O1), 3(O8)}																																																																										
	<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td></td><td></td><td>x</td><td></td><td>x</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>o</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	x			x		x						o							<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td>o</td><td>x</td><td>x</td><td>o</td><td>x</td></tr> <tr><td></td><td></td><td></td><td>x</td><td>o</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>x</td></tr> </table>	x	o	x	x	o	x				x	o							x	11,52 s	AI Win																																				
x			x		x																																																																							
					o																																																																							
x	o	x	x	o	x																																																																							
			x	o																																																																								
					x																																																																							
4.	{1(X0), 2(X1), 3(X6), 4(X5), 5(O8)}	{1(O4), 2(O2), 3(O3), 4(O7)}																																																																										
	<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td></td><td></td><td>x</td><td>x</td><td></td><td>x</td><td>x</td><td>o</td><td>x</td><td>x</td><td>o</td></tr> <tr><td></td><td></td><td></td><td></td><td>o</td><td></td><td></td><td>o</td><td></td><td>o</td><td>o</td><td>x</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td>o</td><td>x</td></tr> </table>	x			x	x		x	x	o	x	x	o					o			o		o	o	x							x			x	o	x	<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td>x</td><td>o</td><td>x</td><td>x</td><td>o</td><td>x</td><td>x</td><td>o</td><td>x</td><td>x</td><td>o</td></tr> <tr><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td></tr> <tr><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td>o</td><td>x</td><td>o</td><td>x</td><td></td></tr> </table>	x	x	o	x	x	o	x	x	o	x	x	o	o	o	x	o	o	x	o	o	x	o	o	x	x			x			x	o	x	o	x		10,61 s	Draw
x			x	x		x	x	o	x	x	o																																																																	
				o			o		o	o	x																																																																	
						x			x	o	x																																																																	
x	x	o	x	x	o	x	x	o	x	x	o																																																																	
o	o	x	o	o	x	o	o	x	o	o	x																																																																	
x			x			x	o	x	o	x																																																																		
5.	{1(X0), 2(X3), 3(X4), 4(X8)}	{1(O2), 2(O6), 3(O5)}																																																																										
	<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td></td><td></td><td>x</td><td></td><td>o</td><td>x</td><td></td><td>o</td><td>x</td><td></td><td>o</td></tr> <tr><td></td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td>x</td><td></td><td>x</td><td>x</td><td>o</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td>o</td><td></td><td></td><td>o</td><td></td><td>x</td></tr> </table>	x			x		o	x		o	x		o				x			x	x		x	x	o							o			o		x	<table border="1" style="background-color: #ffff00;"> <tr><td>x</td><td>o</td><td></td><td>x</td><td>o</td><td></td><td>x</td><td>o</td><td></td><td>x</td><td>o</td><td></td></tr> <tr><td>x</td><td>x</td><td></td><td>x</td><td>x</td><td></td><td>x</td><td>x</td><td></td><td>x</td><td>x</td><td>o</td></tr> <tr><td>o</td><td></td><td></td><td>o</td><td></td><td></td><td>o</td><td></td><td></td><td>o</td><td></td><td>x</td></tr> </table>	x	o		x	o		x	o		x	o		x	x		x	x		x	x		x	x	o	o			o			o			o		x	19,25 s	AI Win
x			x		o	x		o	x		o																																																																	
			x			x	x		x	x	o																																																																	
						o			o		x																																																																	
x	o		x	o		x	o		x	o																																																																		
x	x		x	x		x	x		x	x	o																																																																	
o			o			o			o		x																																																																	
HUMAN VS AI																																																																												
	HUMAN	AI																																																																										
1.	{1(X0), 2(X1), 3(X6), 4(X5), 5(O8)}	{1(O4), 2(O2), 3(O3), 4(O7)}																																																																										

	<table border="1"> <tr><td>X</td><td></td><td></td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td></tr> <tr><td></td><td>O</td><td></td><td></td><td>O</td><td></td><td>O</td><td>O</td><td></td><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td><td>X</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td>O</td><td></td><td>X</td><td>O</td><td>X</td></tr> </table>	X			X	X	O	X	X	O	X	X	O	X	X	O		O			O		O	O		O	O	X	O	O	X							X			X	O		X	O	X	10,67 s	Draw														
X			X	X	O	X	X	O	X	X	O	X	X	O																																																
	O			O		O	O		O	O	X	O	O	X																																																
						X			X	O		X	O	X																																																
2.	<table border="1"> <tr><td colspan="7">{1(X8), 2(X0), 3(X7), 4(X2), 5(O4)}</td><td colspan="7">{1(O4), 2(O1), 3(O6), 3(O5)}</td></tr> <tr><td></td><td></td><td></td><td>X</td><td>O</td><td></td><td>X</td><td>O</td><td></td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td>O</td><td></td><td></td><td>O</td><td></td><td></td><td>O</td><td></td><td></td><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td></tr> <tr><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td></tr> </table>	{1(X8), 2(X0), 3(X7), 4(X2), 5(O4)}							{1(O4), 2(O1), 3(O6), 3(O5)}										X	O		X	O		X	O	X	X	O	X		O			O			O			O	O	X	O	O			X			X	O	X	X	O	X	X	O	X	X	11,55 s	Draw
{1(X8), 2(X0), 3(X7), 4(X2), 5(O4)}							{1(O4), 2(O1), 3(O6), 3(O5)}																																																							
			X	O		X	O		X	O	X	X	O	X																																																
	O			O			O			O	O	X	O	O																																																
		X			X	O	X	X	O	X	X	O	X	X																																																
3.	<table border="1"> <tr><td colspan="7">{1(X1), 2(X7), 3(X8), 4(X2)}</td><td colspan="7">{1(O0), 2(O4), 3(O6), 3(O3)}</td></tr> <tr><td></td><td></td><td></td><td>O</td><td>X</td><td></td><td>O</td><td>X</td><td></td><td>O</td><td>X</td><td></td><td>O</td><td>X</td><td>X</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td><td></td><td></td><td>O</td><td></td><td>O</td><td>O</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td></td><td></td></tr> </table>	{1(X1), 2(X7), 3(X8), 4(X2)}							{1(O0), 2(O4), 3(O6), 3(O3)}										O	X		O	X		O	X		O	X	X								O			O		O	O							X		O	X	X	O	X	X			11,22 s	AI Win
{1(X1), 2(X7), 3(X8), 4(X2)}							{1(O0), 2(O4), 3(O6), 3(O3)}																																																							
			O	X		O	X		O	X		O	X	X																																																
							O			O		O	O																																																	
					X		O	X	X	O	X	X																																																		
4.	<table border="1"> <tr><td colspan="7">{1(X4), 2(X7), 3(X2), 4(X3), 5(O8)}</td><td colspan="7">{1(O0), 2(O1), 3(O6), 3(O5)}</td></tr> <tr><td>O</td><td></td><td></td><td>O</td><td>O</td><td></td><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td><td>X</td></tr> <tr><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td></tr> <tr><td></td><td></td><td></td><td>X</td><td></td><td>O</td><td>X</td><td></td><td></td><td>O</td><td>X</td><td></td><td>O</td><td>X</td><td>X</td></tr> </table>	{1(X4), 2(X7), 3(X2), 4(X3), 5(O8)}							{1(O0), 2(O1), 3(O6), 3(O5)}							O			O	O		O	O	X	O	O	X	O	O	X		X			X			X		X	X	O	X	X	O				X		O	X			O	X		O	X	X	11,18 s	Draw
{1(X4), 2(X7), 3(X2), 4(X3), 5(O8)}							{1(O0), 2(O1), 3(O6), 3(O5)}																																																							
O			O	O		O	O	X	O	O	X	O	O	X																																																
	X			X			X		X	X	O	X	X	O																																																
			X		O	X			O	X		O	X	X																																																
5.	<table border="1"> <tr><td colspan="7">{1(X3), 2(X4), 3(X1), 4(X2), 5(O8)}</td><td colspan="7">{1(O0), 2(O5), 3(O7), 3(O6)}</td></tr> <tr><td>O</td><td></td><td></td><td>O</td><td></td><td></td><td>O</td><td>X</td><td></td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td></tr> <tr><td>X</td><td></td><td></td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td><td>O</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>O</td><td></td><td>O</td><td>O</td><td></td><td>O</td><td>O</td><td>X</td></tr> </table>	{1(X3), 2(X4), 3(X1), 4(X2), 5(O8)}							{1(O0), 2(O5), 3(O7), 3(O6)}							O			O			O	X		O	X	X	O	X	X	X			X	X	O	X	X	O	X	X	O	X	X	O								O		O	O		O	O	X	09,80 s	Draw
{1(X3), 2(X4), 3(X1), 4(X2), 5(O8)}							{1(O0), 2(O5), 3(O7), 3(O6)}																																																							
O			O			O	X		O	X	X	O	X	X																																																
X			X	X	O	X	X	O	X	X	O	X	X	O																																																
							O		O	O		O	O	X																																																

Hasil percobaan permainan tic tac toe dengan AI adalah untuk mengevaluasi efisiensi dan kualitas permainan yang menggunakan algoritma Minimax dan optimasi Alpha-Beta Pruning. Berdasarkan percobaan yang ada pada tabel sebanyak 5 kali untuk AI VS HUMAN dan 5 kali untuk HUMAN VS AI, menggunakan algoritma Minimax dengan optimasi Alpha-Beta Pruning, pertandingan seri sebanyak 50% dan AI menang sebanyak 50%. AI mampu meraih kemenangan sebanyak 80% pada pertandingan AI VS HUMAN, di mana agen berperan sebagai 'X' dan mengambil langkah pertama. Hal tersebut menunjukkan bahwa AI dengan algoritma Minimax dan optimasi Alpha-Beta Pruning mampu memberikan tantangan yang kompetitif bagi pemain manusia. Hasil ini menunjukkan bahwa optimasi Alpha-Beta Pruning berhasil meningkatkan performa AI dalam permainan tic tac toe. Dengan Alpha-Beta Pruning agen mampu menghitung nilai setiap langkah dengan lebih akurat, sehingga dapat memilih langkah yang lebih optimal dan meningkatkan peluangnya untuk menang. Sedangkan peluang manusia untuk menang sangat sedikit karena agen sangat kompetitif.

Hasil

Hasil penelitian menunjukkan bahwa penerapan strategi agen menggunakan kecerdasan buatan (AI) dalam permainan tic tac toe dengan algoritma Minimax dan optimasi Alpha-Beta Pruning cenderung menghasilkan permainan yang seimbang dan kompetitif. AI mampu meraih kemenangan sebanyak 80% pada pertandingan AI VS HUMAN, dimana agen berperan sebagai 'X' dan mengambil langkah pertama setelah itu menyusun strategi untuk melawan manusia. Dalam lingkungan permainan ini, agen AI memiliki peluang untuk meraih kemenangan karena agen secara efektif berusaha untuk memblokir lawan sambil juga mencari peluang untuk memenangkan permainan sendiri. Dengan demikian, permainan cenderung berakhir dengan hasil imbang, menunjukkan bahwa strategi agen telah berhasil menciptakan lingkungan permainan yang menantang dan adil bagi kedua pemain.

KESIMPULAN

Penggunaan algoritma Minimax dengan optimasi Alpha-Beta Pruning dalam permainan tic tac toe telah terbukti meningkatkan efisiensi AI dalam pengambilan keputusan, menghasilkan permainan yang seimbang dan kompetitif di mana kedua pemain memiliki kesempatan yang sama untuk meraih kemenangan. Strategi pemain yang efektif, bersama dengan evaluasi heuristik yang cermat, memainkan peran penting dalam dinamika permainan yang menantang dan strategis. Penggunaan pruning dalam algoritma Minimax memberikan dasar yang kuat untuk pengembangan sistem kecerdasan buatan dalam permainan strategis lainnya, menawarkan potensi untuk meningkatkan pengalaman permainan bagi pengguna secara keseluruhan.

REFERENSI

- A. Christopher, Daniel, E. Pratama, dan L.Hakim, (2020). Penerapan Algoritma Minimax Terhadap Permainan Tic-Tac-Toe Dengan Menggunakan Artificial Intelligence. *Jurnal Teknik Informatika Unika Santo Thomas*, 5(2), pp. 127-136.
- B, S., R, V. and T S R, subashri (2020) 'Analysis of minimax algorithm using tic-tac-toe', *Intelligent Systems and Computer Technology* [Preprint]. doi:10.3233/apc200197.
- Chhikara, P. (2022) *Minimax algorithm explanation using tic-tac-toe game*, Medium. Available at: <https://levelup.gitconnected.com/minimax-algorithm-explanation-using-tic-tac-toe-game-22668694aa13> (Accessed: 20 March 2024).
- Christnatalis, N. P. Sijabat, M. A. Riad, J. S. M. Lumbantobing, dan D. B. Sanjaya, Analisa Efektivitas Algoritma Minimax, Alpha Beta Pruning, dan Negamax dalam Penerapannya pada Permainan Papan (*Board Game*). *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, 3(2), pp. 44-59.
- Dinda Permatasari, B. *et al.* (2022) 'Peningkatan kemenangan non-playable character Dalam Permainan Triple Triad Menggunakan alpha-beta pruning', *Jurnal Komputasi*, 10(1), pp. 95-104.
- Emanuel, G., J. Bendi, R.K. and Arieffianto, A. (2019) 'Desain non-player character Permainan Tic-tac-toe dengan ALGORITMA minimax', *Jurnal Ilmiah Matrik*, 21(3), pp. 223-233.
- Faqih, A.A. and Fitriani, A.S. (2023) *Application of tic-tac-toe game using artificial neural network method in web-based decision making for winning* [Preprint]. doi:10.21070/ups.267.
- Haryanto, D. dan M. N. Huda, (2019). Analisis Performa Algoritma Minimax dan Alpha-Beta Pruning pada Permainan Tic-Tac-Toe. *Jurnal Informatika*, 12(1).
- Hermawan, L. and Ismiati, M.B. (2022) 'Penerapan augmented reality berbasis minimax algorithm pada game Papan Cerdas', *Jurnal Buana Informatika*, 13(1), pp. 21-30.
- Jofanda, A.N. and Yasin, M. (2021) 'Design of checkers game using alpha-beta pruning algorithm', *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, 5(2), pp. 279-295.
- L. I. A. Rinjani, L. A. S. I. Akbar, dan A. S. Rachman, (2020). IMPLEMENTASI MULTITHREADING PADA ALGORITMA MINIMAX. *Jurnal Bakti Nusa*, 1(2), pp. 37-48.
- Pearl, J. (2022) 'The solution for the branching factor of the alpha-beta pruning algorithm and its optimality', *Probabilistic and Causal Inference*, pp. 91-102. doi:10.1145/3501714.3501724.
- R. W. A. Pamungkas dan H. Thamrin, (2022). Penerapan Algoritma Minimax pada Game Tic Tac Toe. *Jurnal Ilmiah Matematika*, 9(2), pp. 82-94.
- Sari, R. A., & Mulyono, B. (2022). Implementasi Permainan Tic Tac Toe Berbasis Android. *Jurnal Informatika dan Komputer Universitas Brawijaya*, 16(1), pp. 1-10.
- Smith, J, (2022). The limitations of traditional tic tac toe: A need for technological innovation. *International Journal of Game Studies*, 24(2), pp. 183-198.
- S. Mandadi, Tejashwini B, dan S. Vijayakumar, (2020). IMPLEMENTATION OF SEQUENTIAL AND PARALLEL ALPHA-BETA PRUNING ALGORITHM. *INTERNATIONAL*

JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT], 7(8), pp. 98-104.

Thamrin, H. and Pamungkas, R.W. (2023) 'Algoritma Minimax Untuk Game Tic Tac Toe Yang Menantang', *Indonesian Journal of Computer Science*, 12(3), pp. 1386-1398.